ALGORITHMS FOR NNLO QCD-EW AND EW CALCULATIONS IN 2 → 2 PROCESSES

Simone Devoto

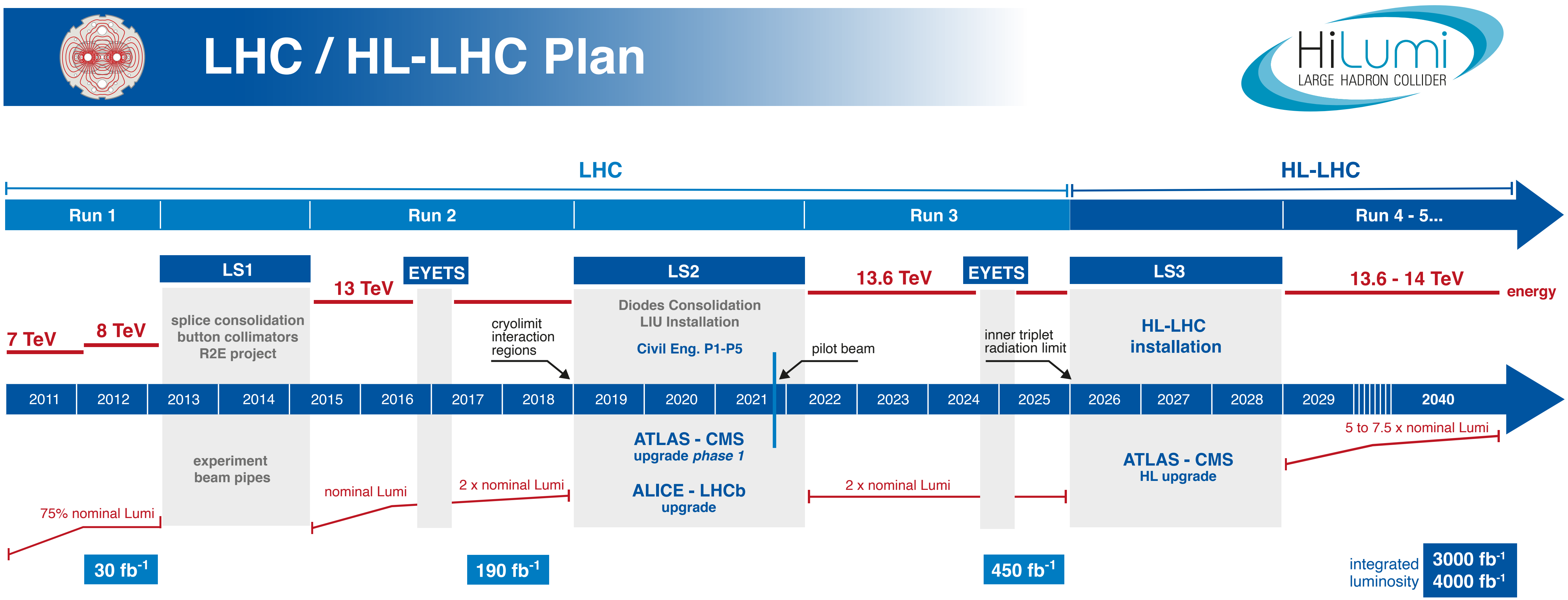
GHENT UNIVERSITY

FACULTY OF SCIENCES

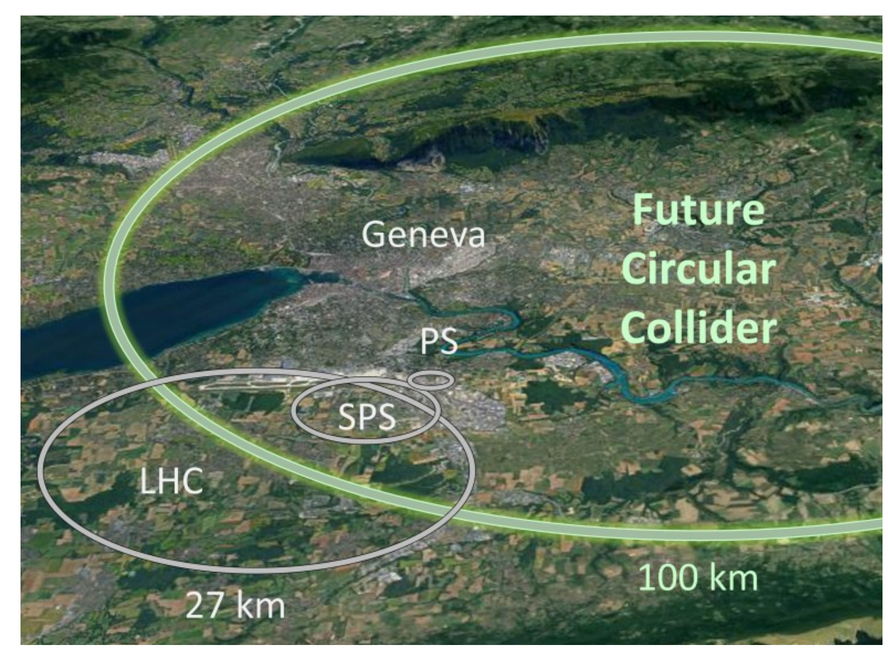
European Research Council
Established by the European Commission

In collaboration with:
T. Armadillo, R. Bonciani, M. Dradi, N. Rana, A. Vicini

THE FUTURE OF LHC

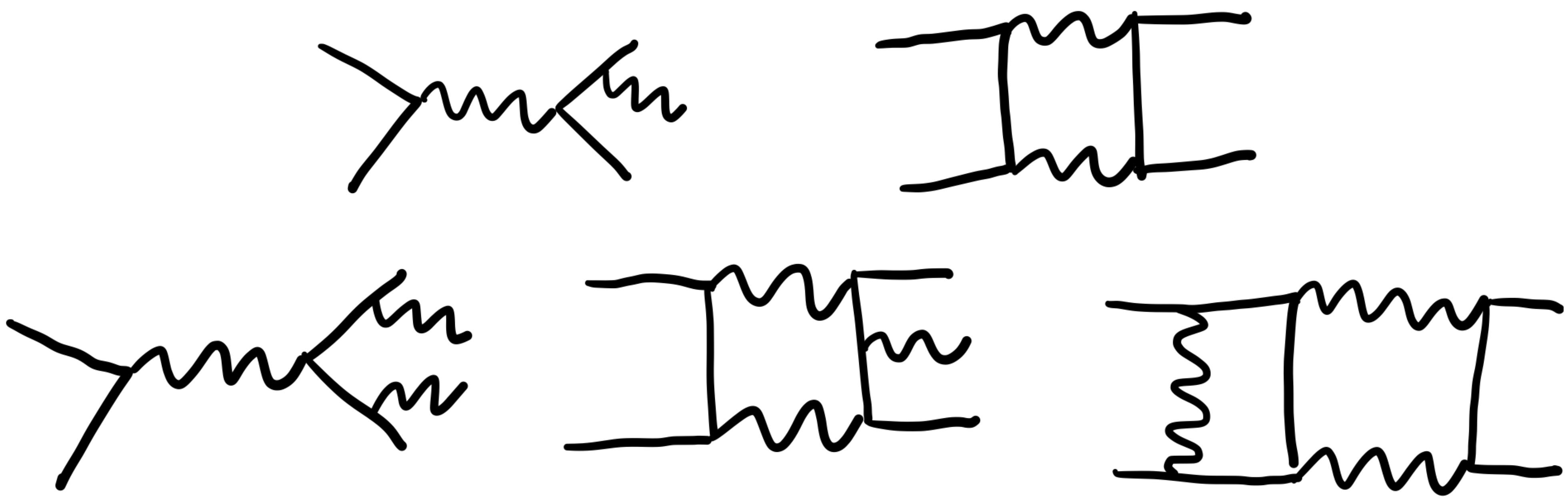


- ➤ **Run3** at LHC: factor of **2** increase of the data set;
- ➤ High Luminosity program (**HL-LHC**): factor of **10** increase of the data set;
- ➤ Dramatic **experimental improvement**, with an expected goal of **1% precision or better** in a key set of observables (**1‰ at FCC**!)



WHY NNLO EW?

- **Theoretical predictions need to match experimental precision.**
- Precision tests of the Standard Model will need the computation of **N3LO QCD**, **NNLO EW** and mixed **NNLO QCDxEW** corrections.
- The computation of **NNLO EW corrections** will be relevant for observables at high invariant masses at **LHC** and will play a crucial role in the study of key processes (e.g. single boson, diboson, top pair proaction…) at future **lepton collider** (e^+e^- phase of FCC?).
- One of the main bottlenecks in the computation of higher order corrections is the evaluation of the required **two loop virtual amplitudes.**



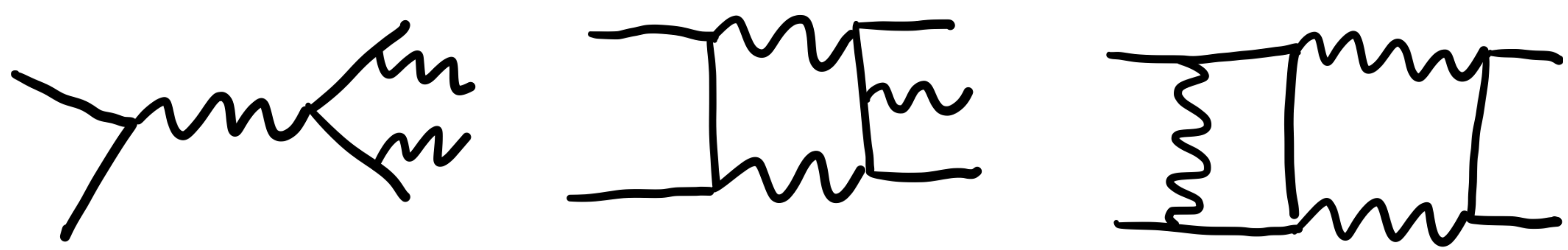
EXTRA CHALLENGES OF NNLO EW

What makes NNLO EW challenging?

- additional internal **massive lines**;
  additional scales in the problem (m_Z, m_W, m_H…)bring additional complications!

- treatment of γ_5;
  how can γ_5 be consistently used in dimensional regularisation?

- need for the **complex mass scheme**;
  requires to analytically continue the master integrals on the complex plane of the kinematical invariants!



CONTENTS

- **Our Workflow**:
  the building blocks;
  - ABISS;
  - SEASYDE;
- **NNLO QCDxEW**:
  a first application;
  - Neutral Current Drell-Yan;
  - Charged Current Drell-Yan;
- **Towards NNLO EW**;
  Future challenges.

OUR WORKFLOW

THE BUILDING BLOCKS

ABISS

Private Mathematica package

SEASYDE

Public Mathematica package

STRUCTURE OF A LOOP COMPUTATION

Process definition

Feynman Amplitudes

Computation of the interference terms

Reduction to a set of Master Integrals

Evaluation of the Master Integrals

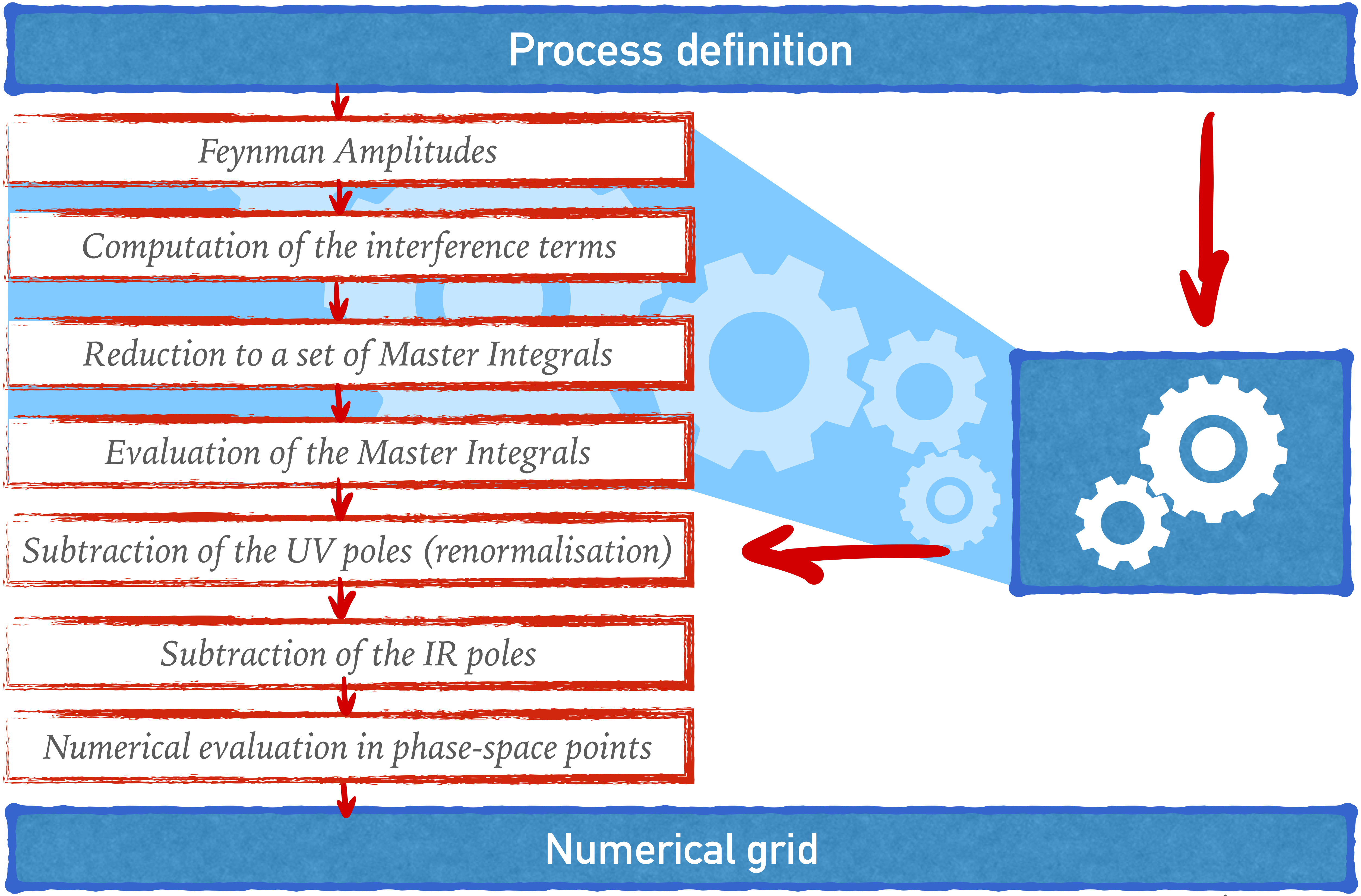
Subtraction of the UV poles (renormalisation)

Subtraction of the IR poles

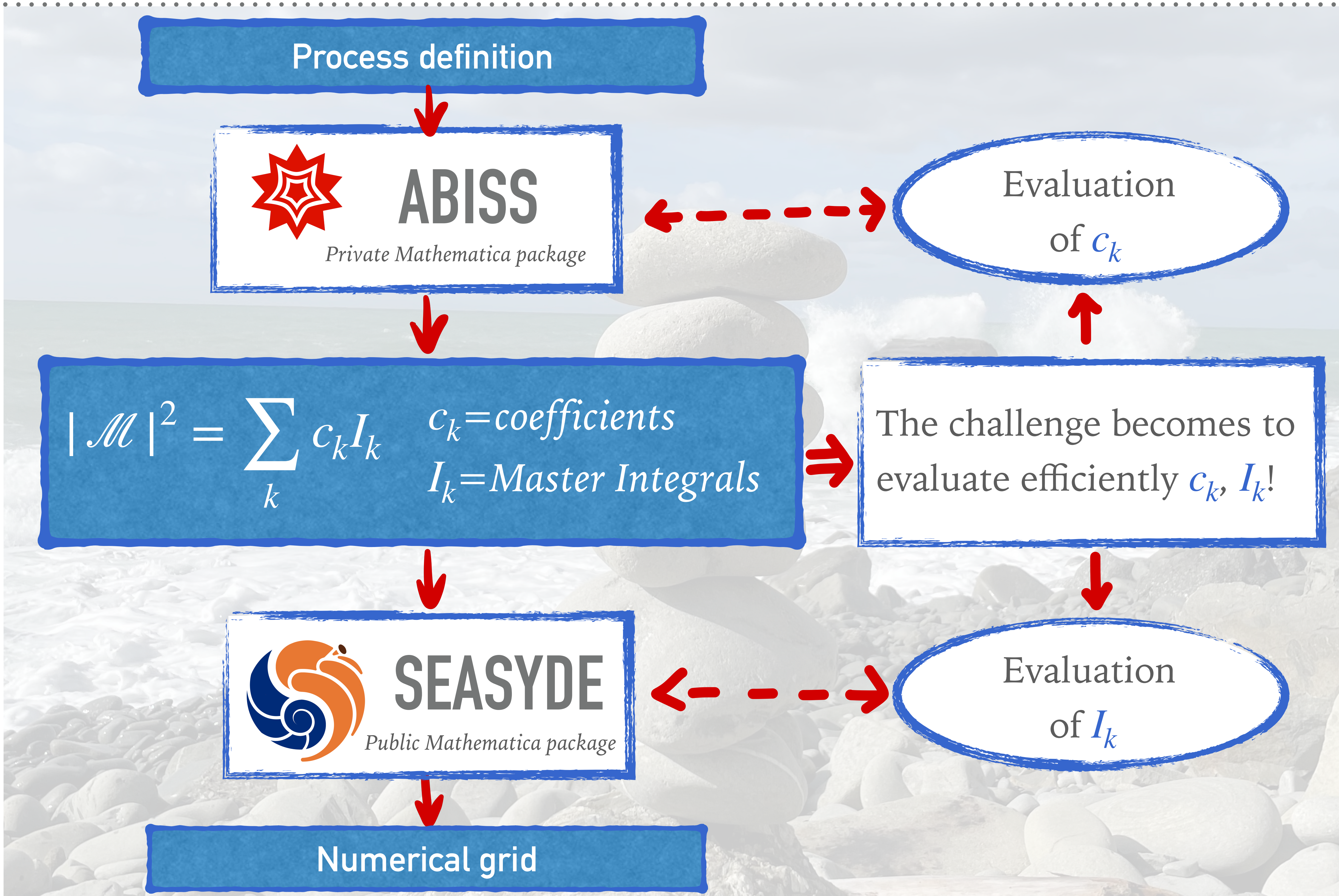
Numerical evaluation in phase-space points

Numerical grid

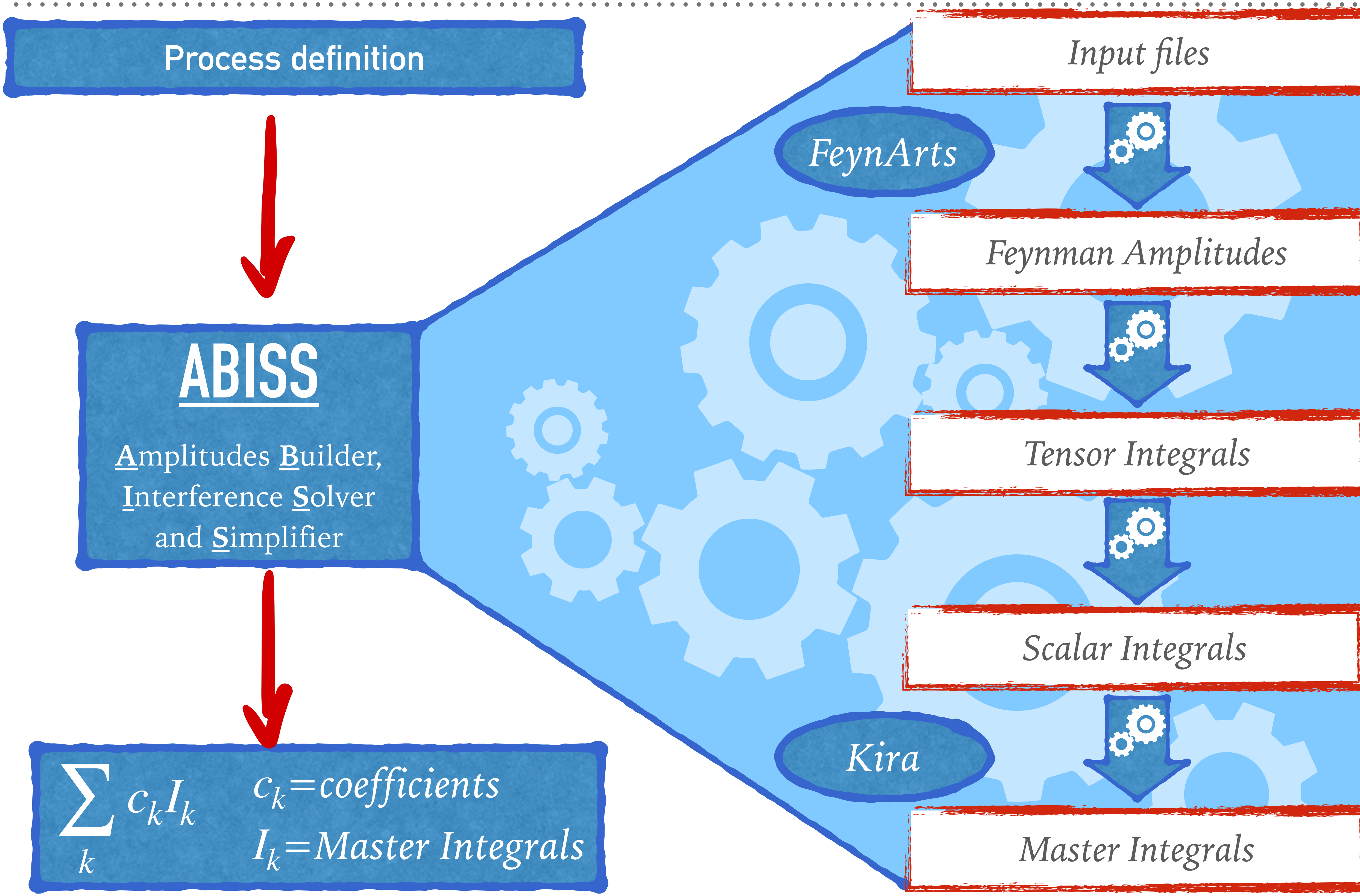
STRUCTURE OF A LOOP COMPUTATION



OUR WORKFLOW



ABISS – EVALUATING c_k



SEASYDE - EVALUATING I_k

[T. Armadillo, R. Bonciani, **SD**, N.Rana, A.Vicini, arXiv:2205.03345]

SEASYDE

The latest version of SEASYDE can be downloaded from: https://github.com/TommasoArmadillo/SeaSyde

- **SEASYDE** (Series Expansion Approach for SYstem of Differential Equations) is a **MATHEMATICA package** for solving the system of differential equation, associated to the Master Integrals of a given topology.
- SEASYDE can handle any **system of coupled differential equations**.
- The method used to solve the system of differential equations is the **series expansion approach**, providing a **semi-analytical solution**.

SEMI-ANALYTICAL?

Numerical Result

Monte Carlo integration or similar techniques.

Analytical Result

The result of the master integral can be expressed in closed form as a combination of elementary and special functions, whose **power expansion and functional relations are known**.

Semi-Analytical Result

The result of the master integral can be expanded as a power series at every point of its domain, but **without** any additional functional relations.

We solve the master integral with **series expansion**!

SOLVING D.E. B SERIES EXPANSION

Method implemented in the Mathematica package DiffExp for real kinematic variables [*F.Moriello, arXiv:1907.13234*], [*M.Hidding, arXiv:2006.05510*] (see also AMFLOW [*X. Liu and Y.-Q. Ma, arXiv: 2201.11669*])

A Simple Example

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

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

$$f'_{hom}(x) = \sum_{k=0}^{\infty} (k+r)\, c_k\, x^{(k+r-1)}$$

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

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

$$f_{part}(x) = f_{hom}(x) \int_0^x dx' \overbrace{\frac{1}{(x'+2)} f_{hom}^{-1}(x')}^{\text{Expanded around } x'=0}$$

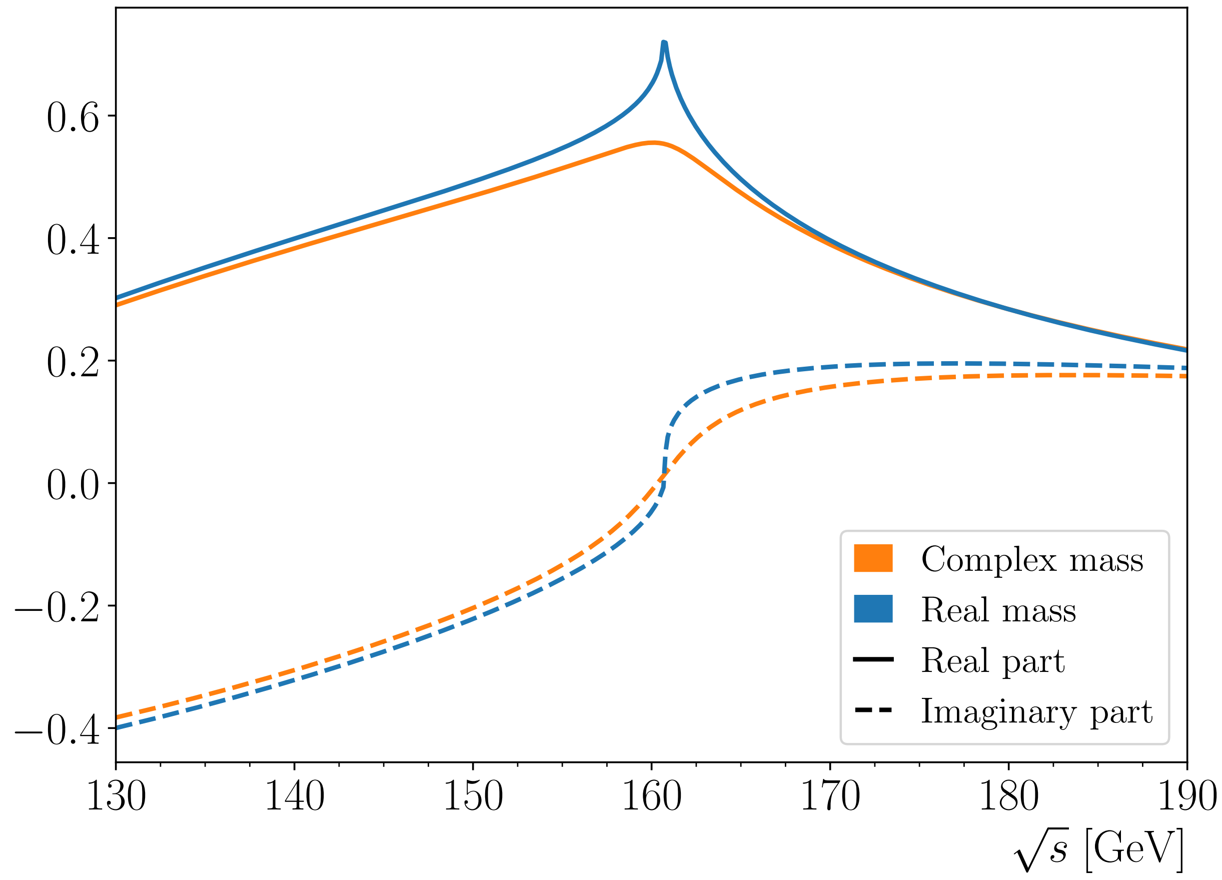
$$= \frac{1}{2}x - \frac{7}{40}x^2 + \frac{2}{75}x^3 + \dots$$

$$f(x) = f_{part}(x) + C f_{hom}(x)$$

$$f(0) = 1 \rightarrow C = \frac{1}{5}$$

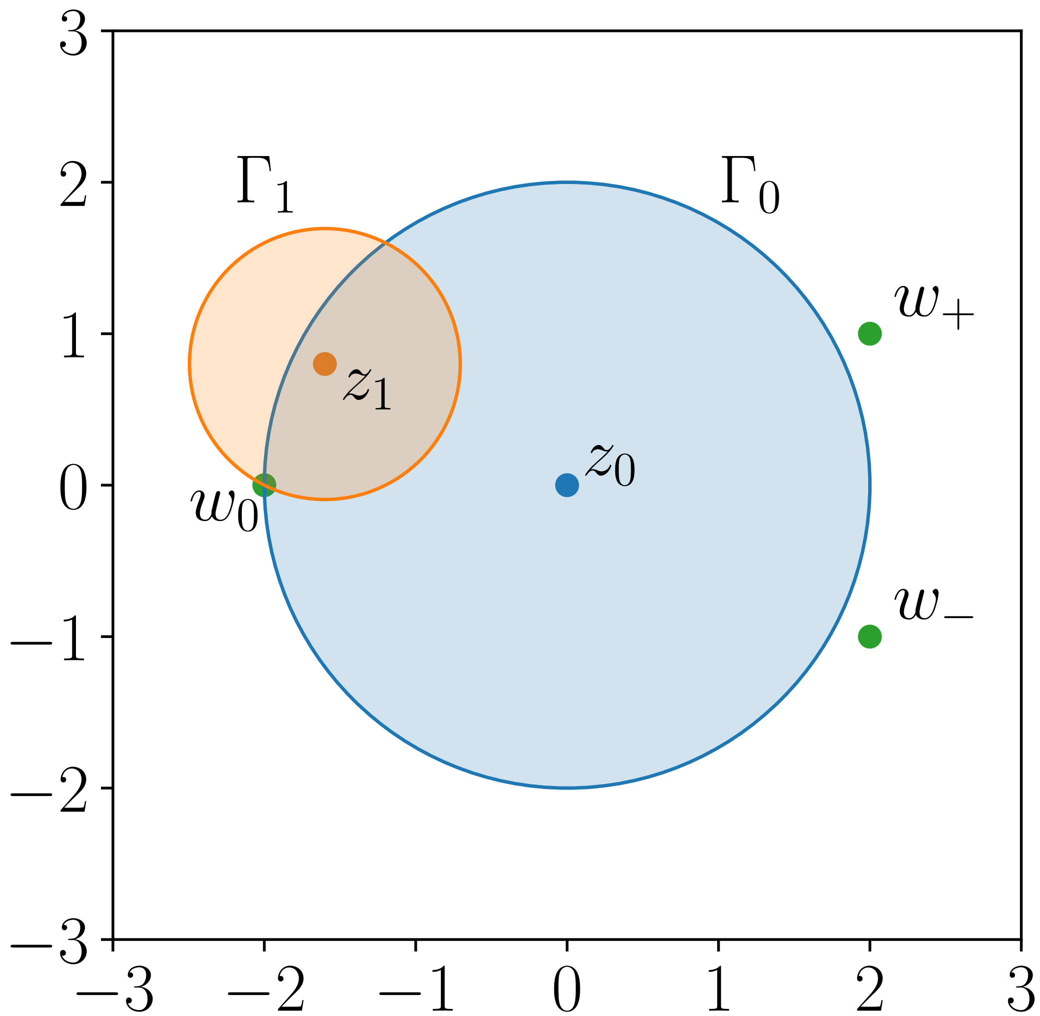
COMPLEX MASS SCHEME

- Starting from NNLO EW, a **Gauge invariant definition** of the mass requires the introduction of the complex mass scheme;
- We introduce the complex mass $\mu_V^2 = m_V^2 - i\Gamma_V m_V$;
- The complex mass scheme **regularise the behaviour at the resonance**: $\frac{1}{s - \mu_V^2 + i\delta}$;
- the **adimensional kinematical variables** become **complex valued**: $\tilde{s} = \frac{s}{m_V^2} \to \frac{s}{\mu_V^2}$.



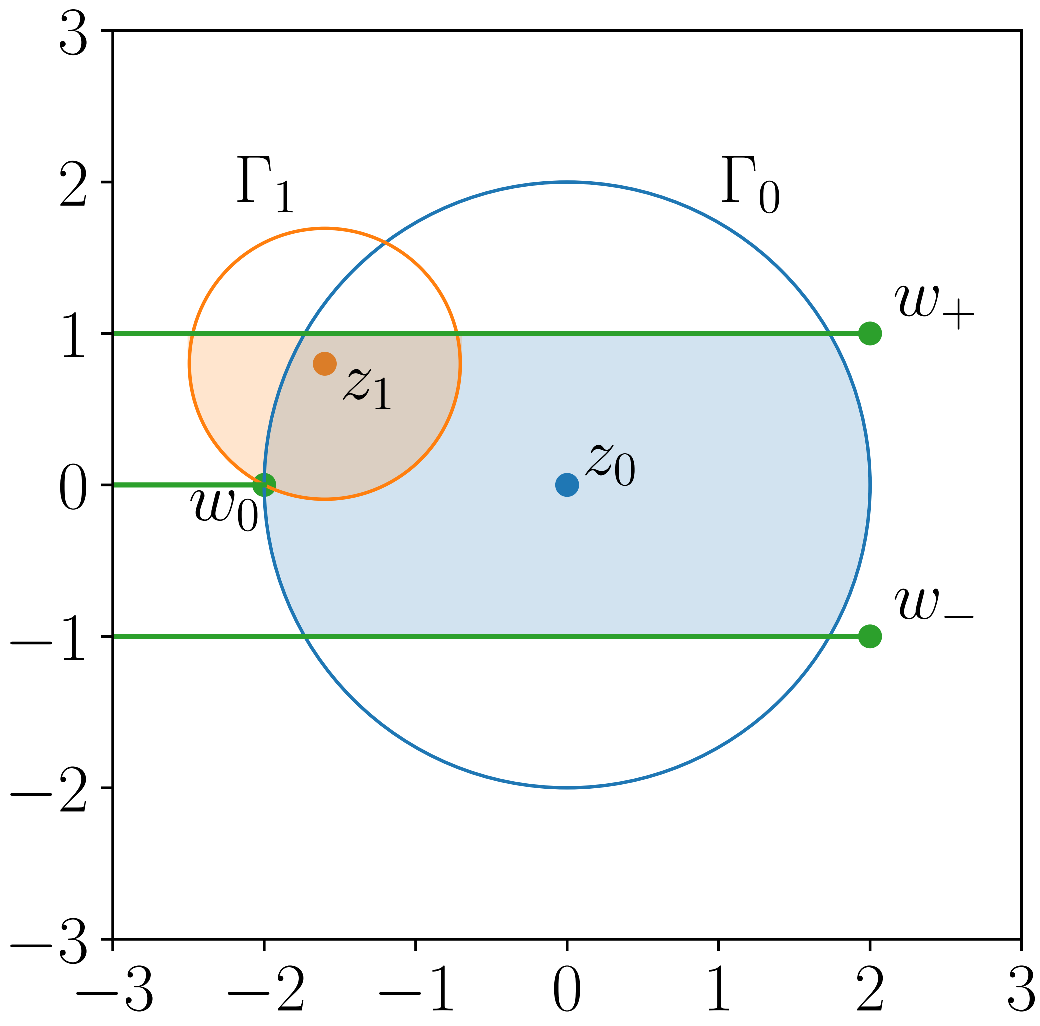
COMPLEX PLANE

- We generalised the series expansion method to **arbitrary complex-valued masses** → **complex plane** of the kinematical invariants!
- The radius of convergence of the series is limited by the presence of **poles**;
- "Transport" of the boundary conditions need to consider **branch-cuts**.



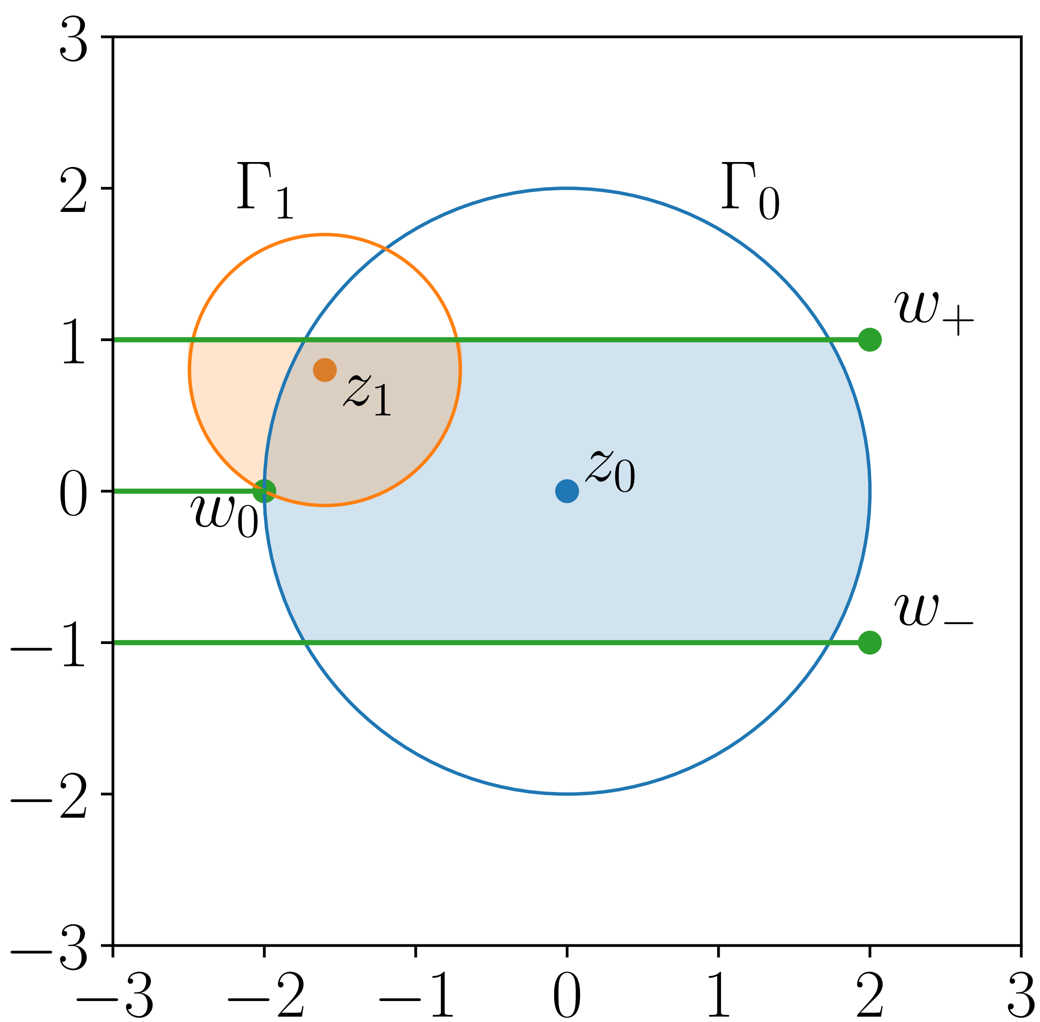
COMPLEX PLANE

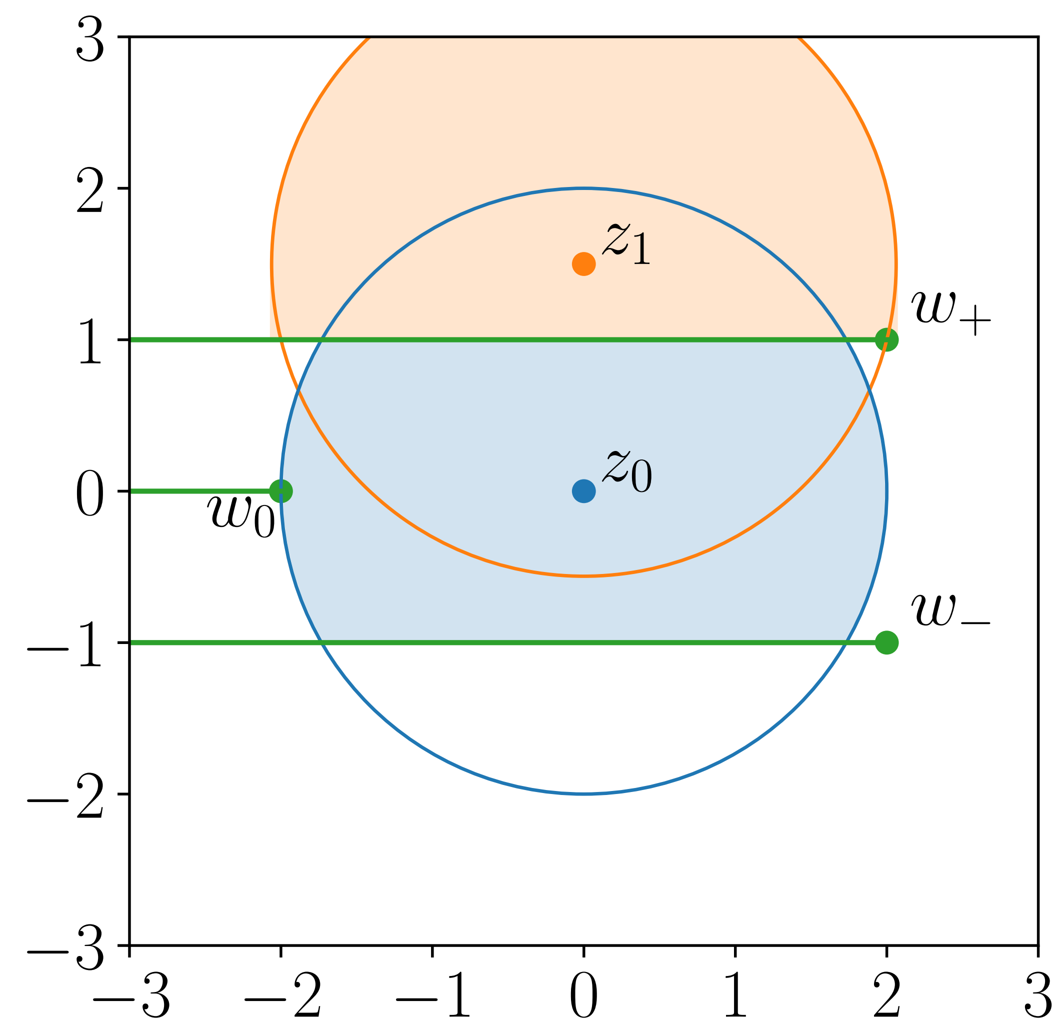
- We generalised the series expansion method to **arbitrary complex-valued masses** ⟶ **complex plane** of the kinematical invariants!
- The radius of convergence of the series is limited by the presence of **poles**;
- "Transport" of the boundary conditions need to consider **branch-cuts**.



COMPLEX PLANE

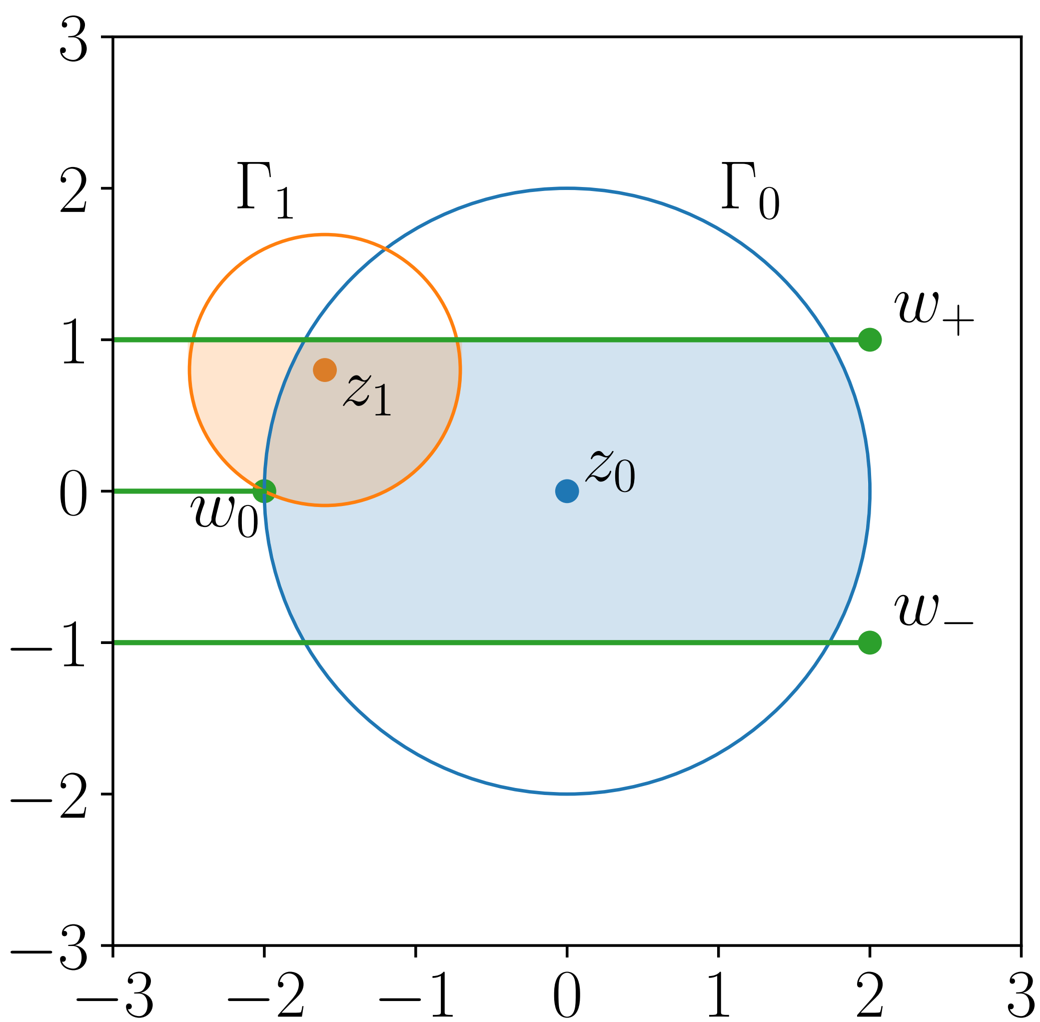
- We generalised the series expansion method to **arbitrary complex-valued masses** → **complex plane** of the kinematical invariants!
- The radius of convergence of the series is limited by the presence of **poles**;
- "Transport" of the boundary conditions need to consider **branch-cuts**.

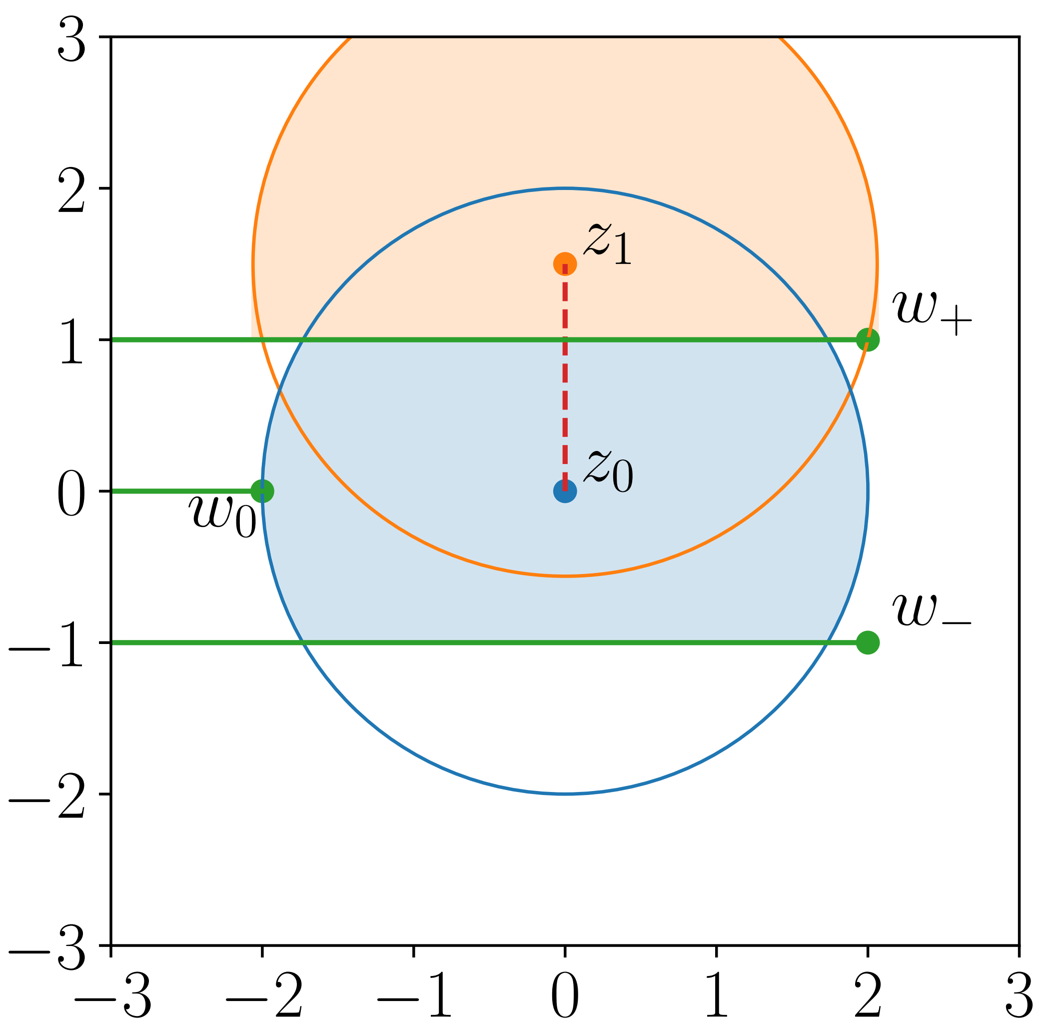




COMPLEX PLANE

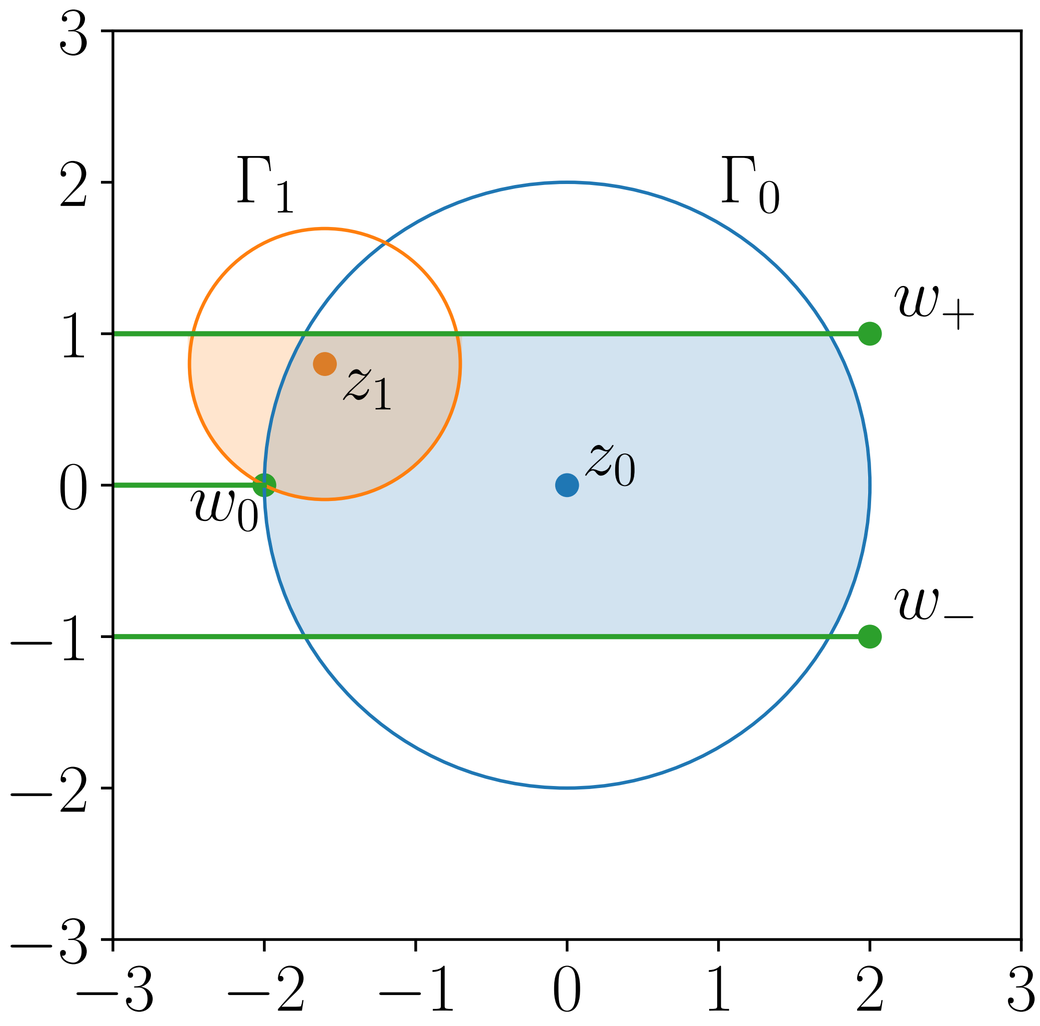
- We generalised the series expansion method to **arbitrary complex-valued masses** → **complex plane** of the kinematical invariants!
- The radius of convergence of the series is limited by the presence of **poles**;
- “Transport” of the boundary conditions need to consider **branch-cuts**.

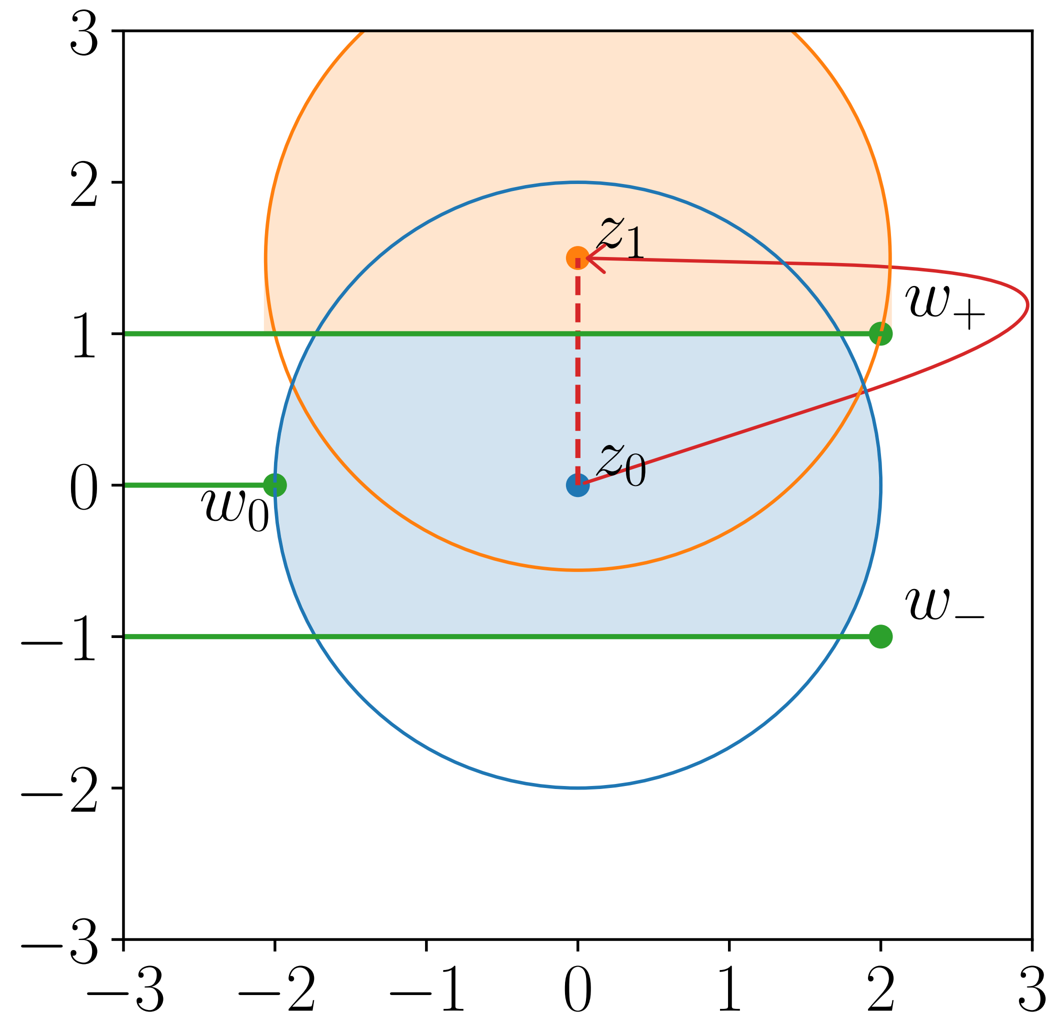




COMPLEX PLANE

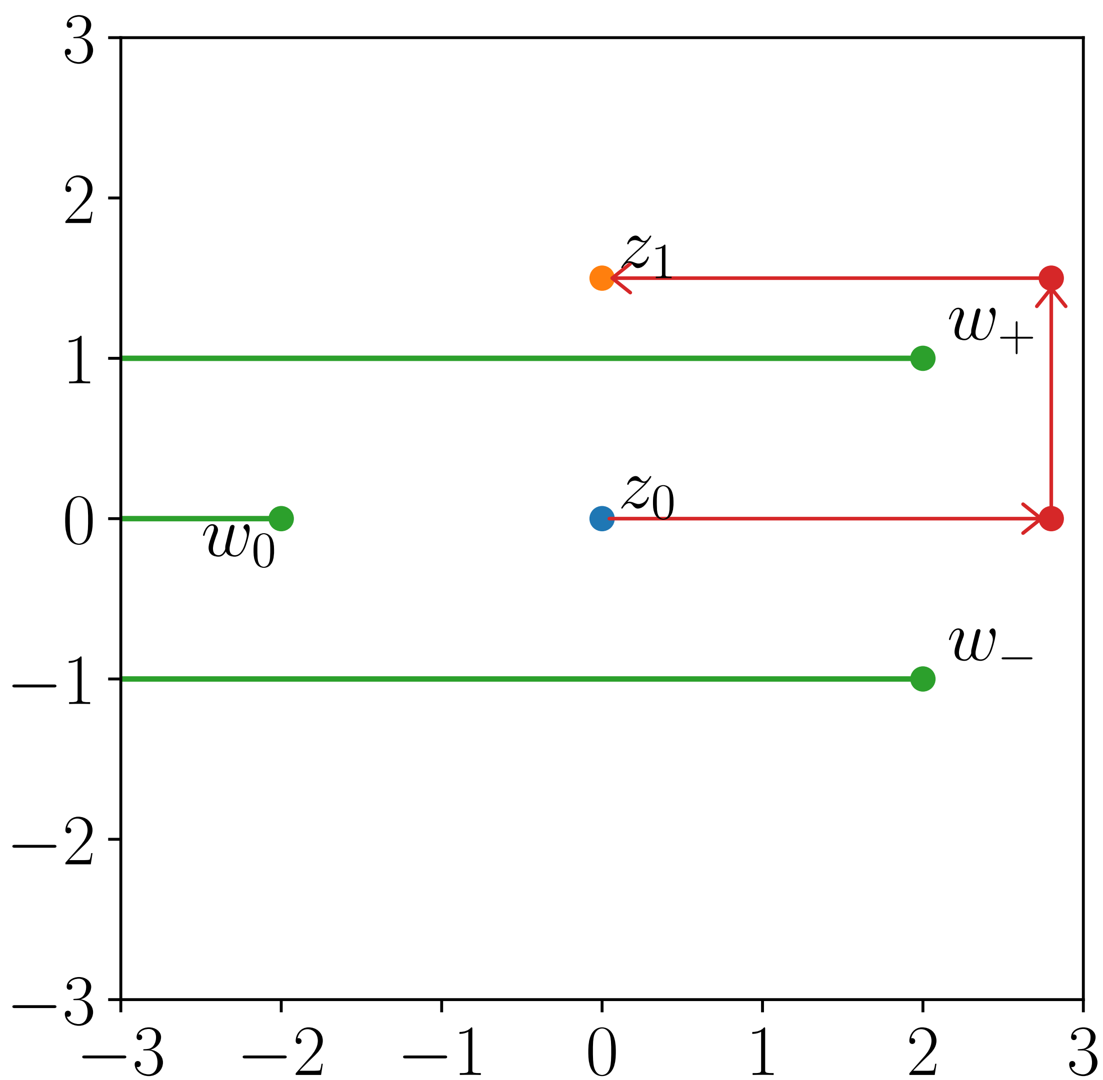
- We generalised the series expansion method to **arbitrary complex-valued masses** ⟶ **complex plane** of the kinematical invariants!
- The radius of convergence of the series is limited by the presence of **poles**;
- "Transport" of the boundary conditions need to consider **branch-cuts**.

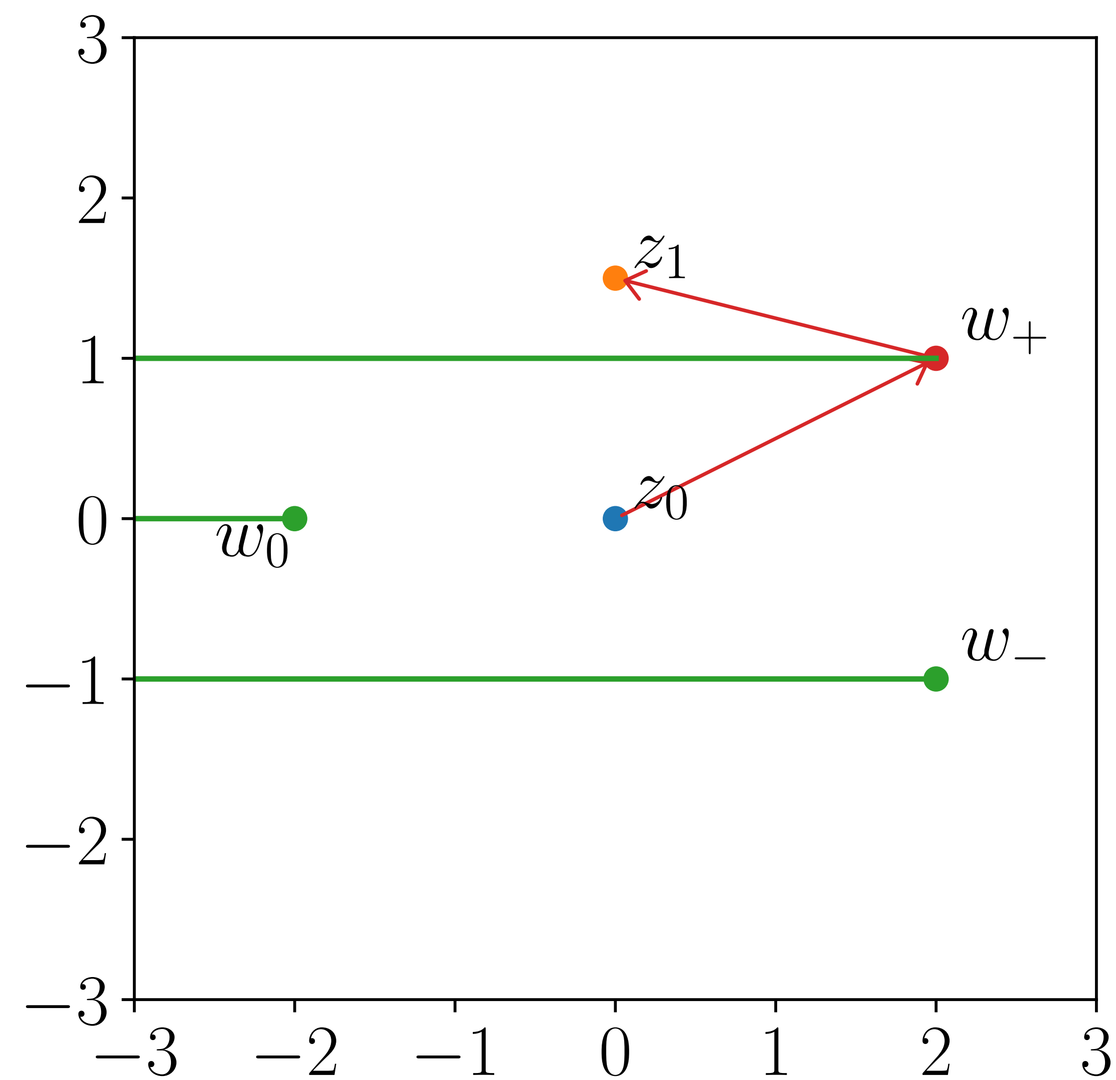




TAYLOR VS LOGARITHMIC EXPANSION

- **Taylor expansion**: **avoids** the singularities;
- **Logarithmic expansion**: uses the singularities as **expansion points**.
- Logarithmic expansion has larger convergence radius but requires longer evaluation time. **We use Taylor expansion as default**.





AMFLOW

[X. Liu, Y. Ma, arXiv:2201.11669]

AMFLOW

- Mathematica package that independently implements the series expansion method;
- by using the **auxiliary mass flow** method automatically obtains the **boundary conditions** of any master integral:

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

the **auxiliary integral** is analytically solved in the limit $\eta \to \infty$ and then evolved to the physical value with the differential equation in η.

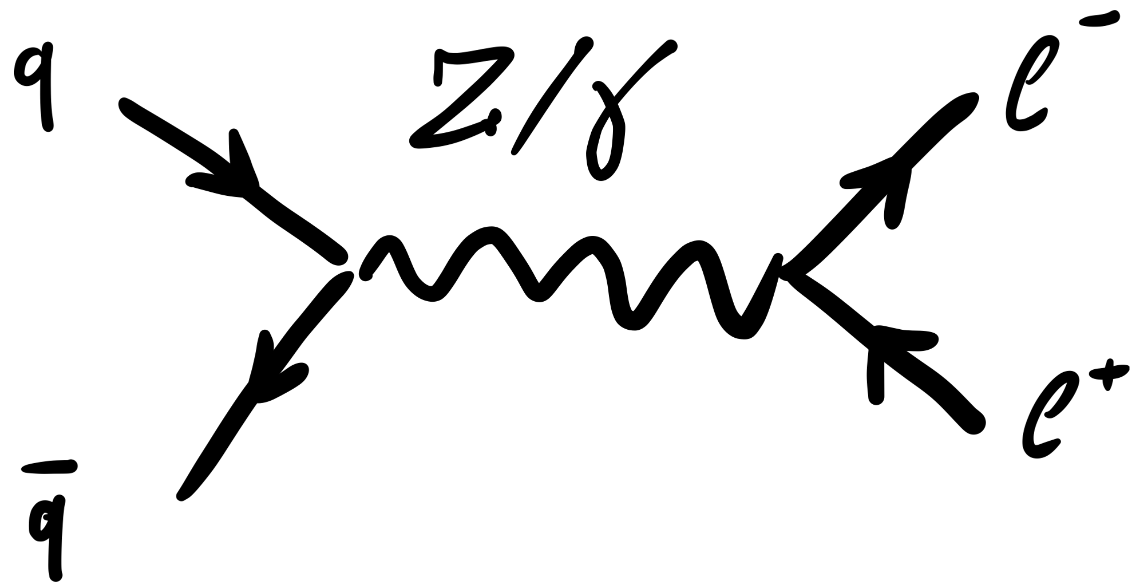
- It is an important tool at our disposal: it guarantees we can <u>**always**</u> find the boundary condition for our system of differential equations!

NNLO QCDxEW

A FIRST APPLICATION

MIXED QCDxEW CORRECTIONS

$$q(p_1) + \bar{q}(p_2) \to l^-(p_3) + l^+(p_4) \quad = \quad \sigma^{(0,0)}$$ *Drell-Yan (1970)*



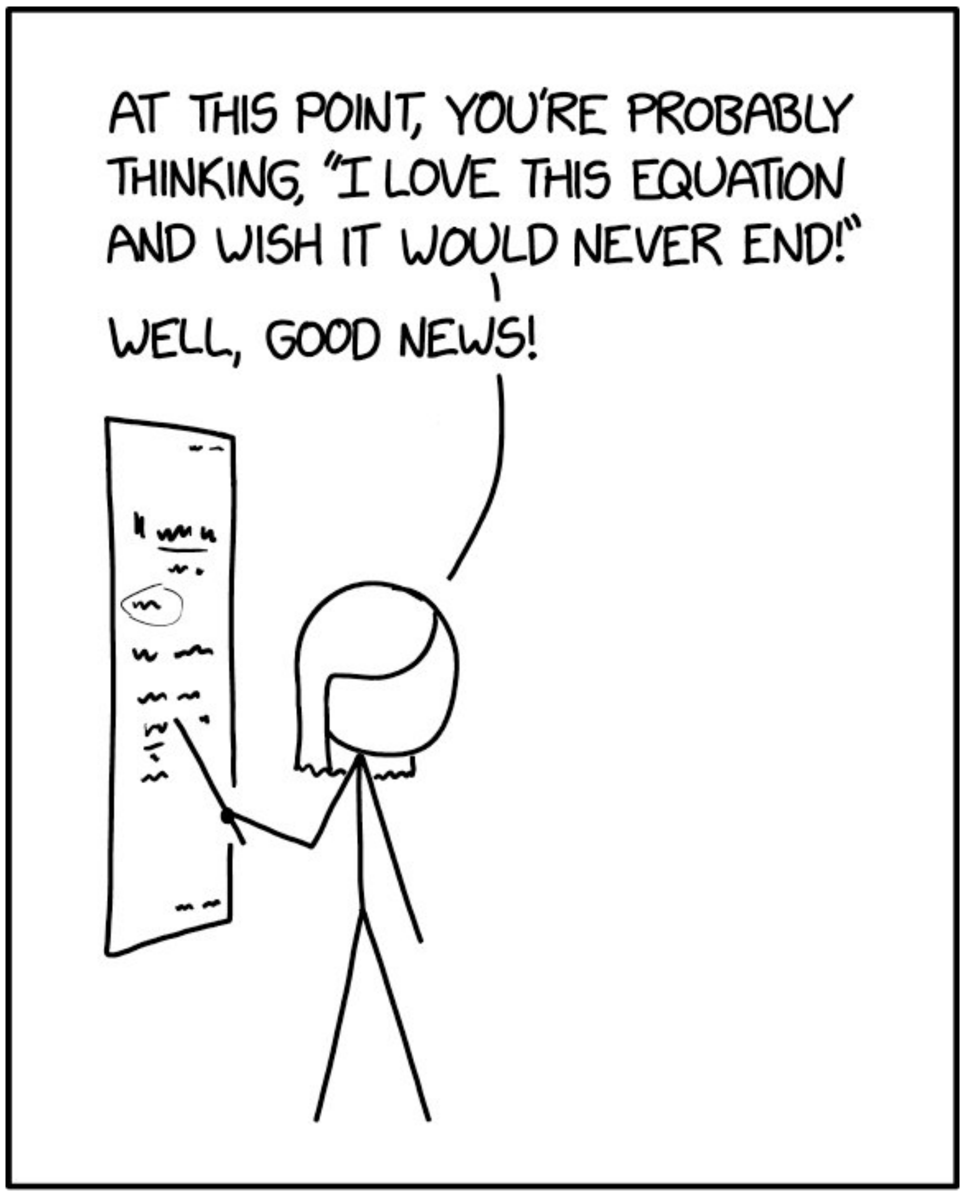
MIXED QCDxEW CORRECTIONS

$$q(p_1) + \bar{q}(p_2) \to l^-(p_3) + l^+(p_4) \quad = \quad \sigma^{(0,0)}$$ *Drell-Yan (1970)*

$$+ \alpha_S\, \sigma^{(1,0)} \quad + \alpha\, \sigma^{(0,1)}$$

$$+ \alpha_S^2\, \sigma^{(2,0)} \quad + \alpha\, \alpha_S\, \sigma^{(1,1)} \quad + \alpha^2\, \sigma^{(0,2)}$$

$$+ \alpha_S^3\, \sigma^{(3,0)} \quad + \ldots$$



TAYLOR SERIES EXPANSION IS THE WORST.

credits: xkcd (2605)

QCD **MIXED** **EW**

MIXED QCDxEW CORRECTIONS

$$q(p_1) + \bar{q}(p_2) \rightarrow l^-(p_3) + l^+(p_4) \quad = \quad \sigma^{(0,0)}$$

QCD CORRECTIONS

$$+ \alpha_S \, \sigma^{(1,0)} + \alpha \, \sigma^{(0,1)}$$

$$+ \alpha_S^2 \, \sigma^{(2,0)} + \alpha \, \alpha_S \, \sigma^{(1,1)} + \alpha^2 \, \sigma^{(0,2)}$$

$$+ \alpha_S^3 \, \sigma^{(3,0)} + \ldots$$

NLO:
[G.Altarelli, R.Ellis, G.Martinelli Nucl.Phys.B 157 (1979)];

NNLO:
[R.Hamberg, T.Matsuura, W.van Nerveen, Nucl. Phys. B 359 (1991)];
[C.Anastasiou, L.J.Dixon, K.Melnikov, F.Petriello, hep-ph:0306192];
[S.Catani, L.Cieri, G.Ferrera, D.de Florian, M.Grazzini arXiv:0903.2120];

N3LO:
[C.Duhr, F.Dulat, B.Mistlberger arXiv:2007.13313];
[X.Chen, T.Gehrmann, N.Glover, A.Huss, T.Yang, and H.Zhu arXiv:2107.09085];
[S.Camarda, L.Cieri, G.Ferrera arXiv:2103.04974];
[X.Chen, T.Gehrmann, N.Glover, A.Huss, P.Monni, E.Re, L.Rottoli, P.Torrielli arXiv:2203.01565];
[T.Neumann, J.Campbell arXiv:2207.07056]

MIXED QCDxEW CORRECTIONS

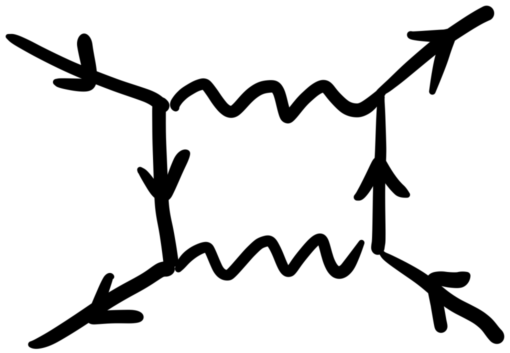
$$q(p_1) + \bar{q}(p_2) \to l^-(p_3) + l^+(p_4) \quad = \quad \sigma^{(0,0)}$$

$$+ \alpha_S\, \sigma^{(1,0)} + \alpha\, \sigma^{(0,1)}$$

$$+ \alpha_S^2\, \sigma^{(2,0)} + \alpha\, \alpha_S\, \sigma^{(1,1)} + \alpha^2\, \sigma^{(0,2)}$$

$$+ \alpha_S^3\, \sigma^{(3,0)} + \ldots$$

EW CORRECTIONS



- NLO corrections known;

  [U.Baur, O.Brein, W.Hollik, C.Schappacher, D.Wackeroth, hep-ph:0108274];
  [S.Dittmaier, M.Kramer, hep-ph:0109062];
  [U.Baur, D.Wackeroth, hep-ph:0405191];

- NNLO corrections still missing (available Sudakov high energy approximation).

  [B. Jantzen, J.H.Kühn. A.A.Penin, V.A.Smirnov, hep-ph:0509157];

MIXED QCDxEW CORRECTIONS

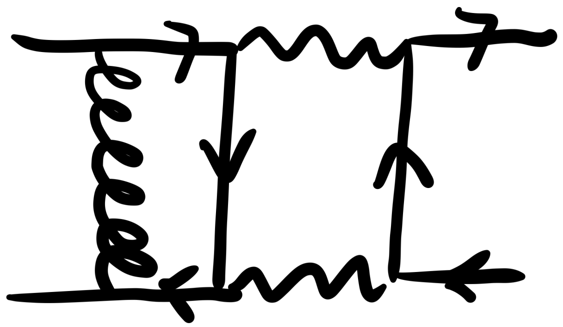
$$q(p_1) + \bar{q}(p_2) \to l^-(p_3) + l^+(p_4) \quad = \quad \sigma^{(0,0)}$$

$$+ \alpha_S\, \sigma^{(1,0)} + \alpha\, \sigma^{(0,1)}$$

$$+ \alpha_S^2\, \sigma^{(2,0)} + \alpha\, \alpha_S\, \sigma^{(1,1)} + \alpha^2\, \sigma^{(0,2)}$$

$$+ \alpha_S^3\, \sigma^{(3,0)} + \ldots$$

MIXED CORRECTIONS



- Recently computed

  [R.Bonciani, L.Buonocore, M.Grazzini, S.Kallweit, N.Rana, F.Tramontano, A.Vicini, arXiv:2106.11953]

  [F.Buccioni, F.Caola, H.Chawdhry, F.Devoto, M.Heller, A.von Manteuffel, K.Melnikov, R.Röntsch, C.Signorile-Signorile, arXiv:2203.11237]

RECENT PROGRESSES IN MIXED CORRECTIONS

- **Theoretical Developments**
  - 2-loop virtual Master Integrals with internal masses *[U. Aglietti, R. Bonciani, arXiv:0304028, arXiv:0401193],[R. Bonciani, S. Di Vita, P. Mastrolia, U. Schubert, arXiv:1604.08581], [M.Heller, A.von Manteuffel, R.Schabinger arXiv:1907.00491], [M.Long,R,Zhang,W.Ma,Y,Jiang,L.Han,,Z.Li,S.Wang, arXiv:2111.14130],[X.Liu, Y.Ma, arXiv:2201.11669]*
  - Altarelli-Parisi splitting functions including QCD-QED effects *[D. de Florian, G. Sborlini, G. Rodrigo, arXiv:1512.00612]*
  - Renormalisation *[G.Degrassi, A.Vicini, hep-ph/0307122],[S.Dittmaier,T.Schmidt,J.Schwarz, arXiv:2009.02229], [S.Dittmaier, arXiv:2101.05154]*
- **On-shell Z and W production**
  - pole approximation of the NNLO QCD-EW corrections *[S.Dittmaier, A.Huss, C.Schwinn, arXiv:1403.3216, 1511.08016]*
  - analytical total Z production cross section including NNLO QCD-QED corrections *[D. de Florian, M.Der, I.Fabre, arXiv:1805.12214]*
  - fully differential on-shell Z production including exact NNLO QCD-QED corrections *[M.Delto, M.Jaquier, K.Melnikov, R.Roentsch, arXiv:1909.08428] [S.Hasan, U.Schubert, arXiv:2004.14908]*
  - analytical total Z production cross section including NNLO QCD-EW corrections *[R. Bonciani, F. Buccioni, R.Mondini, A.Vicini, arXiv:1611.00645], [R. Bonciani, F. Buccioni, N.Rana, I.Triscari, A.Vicini, arXiv:1911.06200], [R. Bonciani, F. Buccioni, N.Rana, A.Vicini, arXiv:2007.06518, arXiv:2111.12694]*
  - fully differential Z and W production including NNLO QCD-EW corrections *[F. Buccioni, F. Caola, M.Delto, M.Jaquier, K.Melnikov, R.Roentsch, arXiv:2005.10221], [A. Behring, F. Buccioni, F. Caola, M.Delto, M.Jaquier, K.Melnikov, R.Roentsch, arXiv:2009.10386, 2103.02671]*
- **Complete Drell-Yan**
  - neutrino-pair production including NNLO QCD-QED corrections *[L. Cieri, D. de Florian, M.Der, J.Mazzitelli, arXiv:2005.01315]*
  - 2-loop amplitudes *[M.Heller, A.von Manteuffel, R.Schabinger, arXiv:2012.05918],[T.Armadillo, R.Bonciani, SD, N.Rana, A.Vicini, arXiv:2201.01754]*
  - NNLO QCD-EW corrections to neutral-current DY including leptonic decay *[R.Bonciani, L.Buonocore, M.Grazzini, S.Kallweit, N.Rana, F.Tramontano, A.Vicini, arXiv:2106.11953],[F.Buccioni, F.Caola, H.Chawdhry, F.Devoto, M.Heller,A.von Manteuffel, K.Melnikov, R.Röntsch, C.Signorile-Signorile, arXiv:2203.11237]*
  - NNLO QCD-EW corrections to charged-current DY including leptonic decay (2-loop contributions in pole approximation). *[L.Buonocore, M.Grazzini, S.Kallweit, C.Savoini, F.Tramontano, arXiv:2102.12539]*

COMPUTATIONAL FRAMEWORK

[T. Armadillo, R. Bonciani, **SD**, N.Rana, A.Vicini, arXiv:2201.01754]

- IR singularities handled by **q_T-subtraction formalism**; [S. Catani, M. Grazzini (2007)] [L.Buonocore, M. Grazzini, F.Tramontano (2019)]

$$d\sigma^{F}_{(N)NLO} = \mathscr{H}^{F}_{(N)NLO} \otimes d\sigma^{F}_{LO} + \left[d\sigma^{F+jets}_{(N)LO} - d\sigma^{CT}_{(N)LO} \right]$$

- straightforward implementation of **any other framework** by replacing the **subtraction operator**.

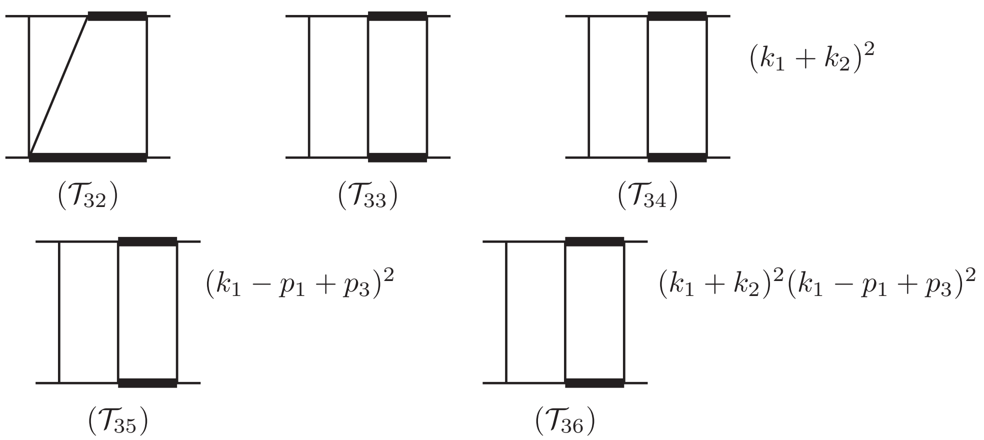
- Final-state collinear singularities regularised by the **lepton mass**;
- **small lepton mass limit**: consider the ratio $m_l/\sqrt{s}$ and keep only **logarithmic terms** $\sim \log\left(m_l/\sqrt{s}\right)$;

- When dealing with intermediate unstable particles, such as W and Z, it is useful to perform the calculations in the **complex-mass scheme**;
- We introduce the complex mass $\mu_V^2 = m_V^2 - i\Gamma_V m_V$ for both the Z and W bosons.

BASIS OF MASTER INTEGRALS

Basis of Master integrals composed by:

- MIs relevant for the **QCD-QED corrections**, with massive final state;
  [R.Bonciani, A.Ferroglia, T.Gehrmann, D.Maitre, C.Studerus, arXiv:0806.2301, 0906.3671]
- MIs with 1or 2 internal mass relevant for the **EW form factor**;
  [U.Aglietti, R.Bonciani, hep-ph/0304028, hep-ph/0401193]
- 31 MIs with 1 mass and **36 MIs with 2 masses** including boxes, relevant for the **QCD-EW corrections to the full Drell-Yan**.
- 5 box integrals are in **Chen-Goncharov** representation; *[R. Bonciani, S. Di Vita, P. Mastrolia, U. Schubert, arXiv:1604.08581]*

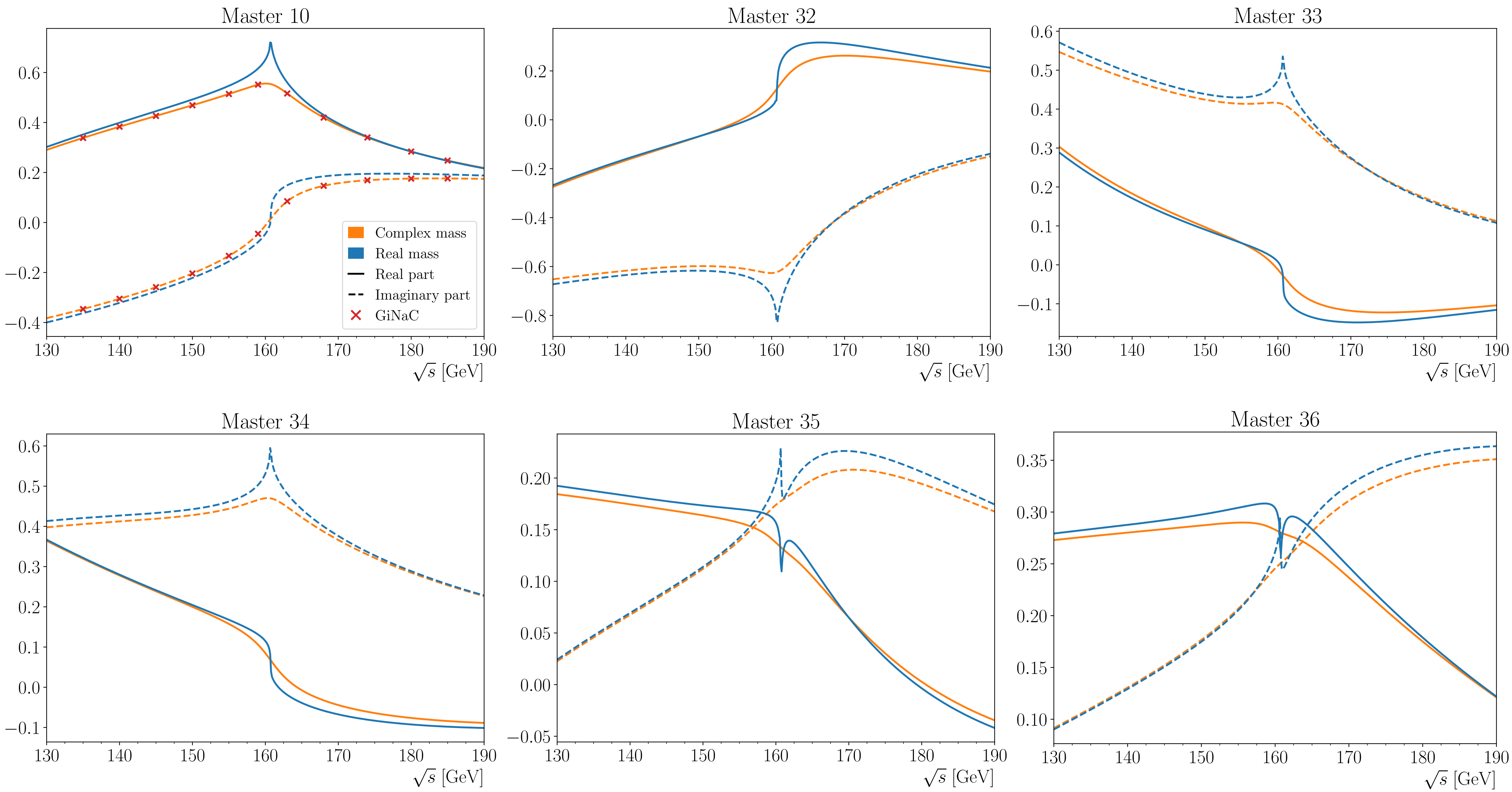


  - difficult numerical evaluation requires alternative strategy.
  - closed form in terms of GPLs found but not public *[M.Heller, A.von Manteuffel, R.M. Schabinger, arXiv:1907.00491]*

NUMERICAL GRIDS (MASTERS)

*[T. Armadillo, R. Bonciani, **SD**, N.Rana, A.Vicini, arXiv:2205.03345]*

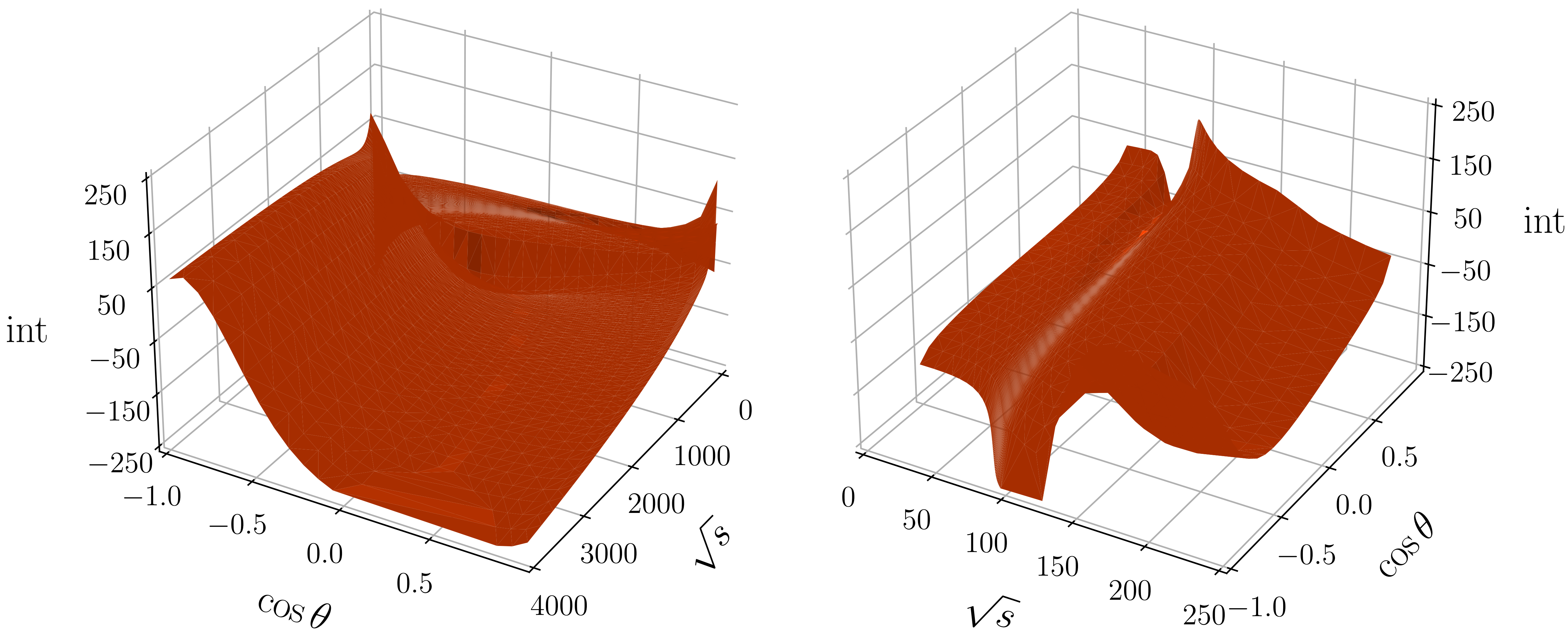
- 31 out of 36 masters known in terms of GPLs: **validation** of SEASYDE.
- 5 out of 36 masters are a genuine SEASYDE **prediction**;
- solution can be computed with **arbitrary number of significant digits**.



NUMERICAL GRIDS

*[T. Armadillo, R. Bonciani, **SD**, N.Rana, A.Vicini, arXiv:2201.01754]*

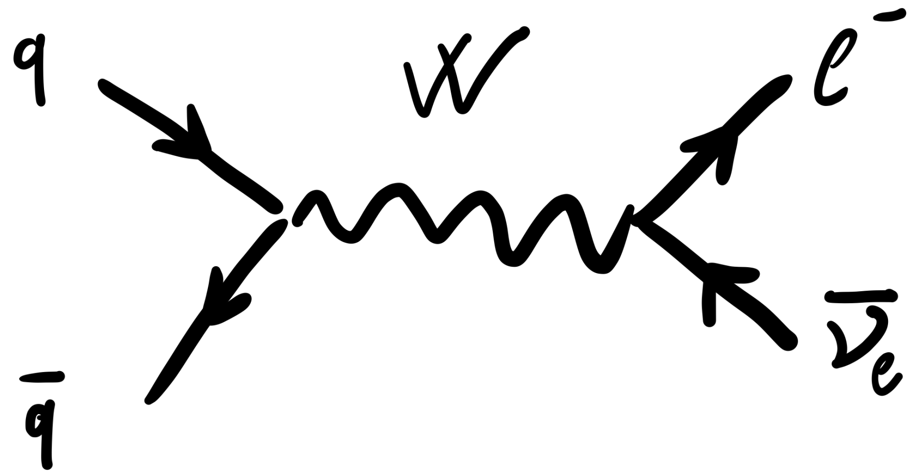
- After subtracting IR and UV divergences, we obtain the **hard function**;
- **Publicly available** as a MATHEMATICA notebook;
- Subtraction of the IR poles done in the **qT-subtraction** formalism;
- Production of the grid (3250 points) required $\mathcal{O}(12\text{h})$ on a 32-cores machine;
- Interpolation of the grid with excellent accuracy requires **negligible time**.



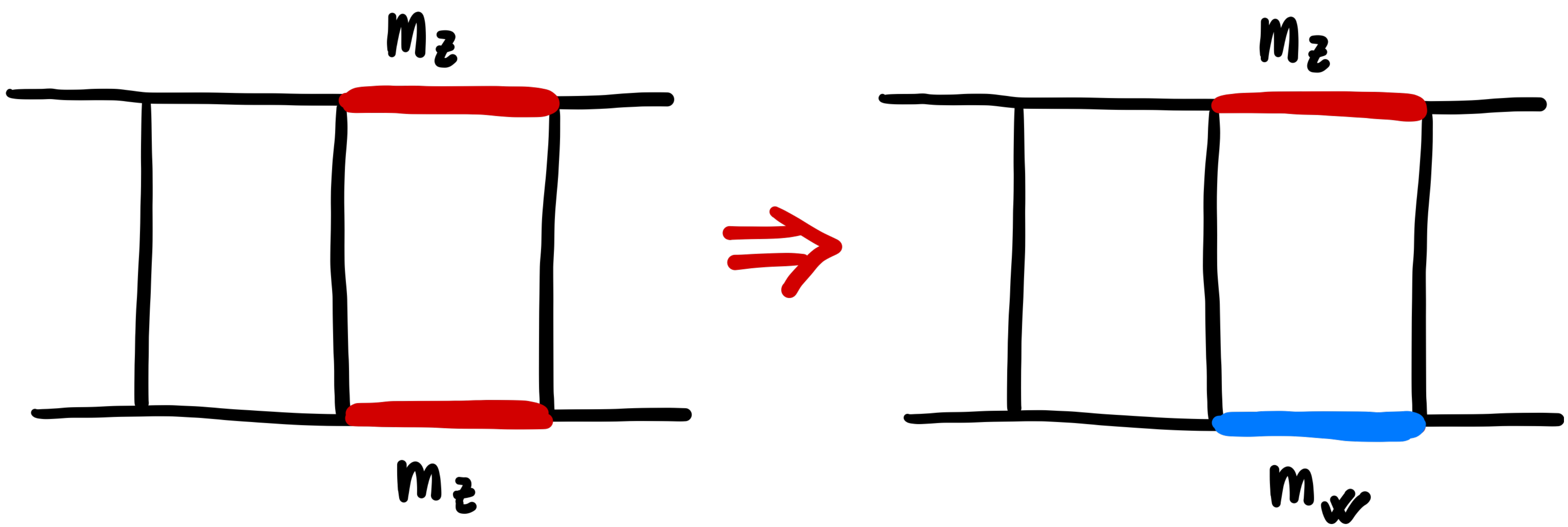
CHARGED CURRENT DRELL-YAN

*[T. Armadillo, R. Bonciani, **SD**, N.Rana, A.Vicini]*

-Work in progress-



- Computationally **similar** to neutral current Drell-Yan;
- Extra complexity coming from new diagrams where **two different internal massive lines** appear:

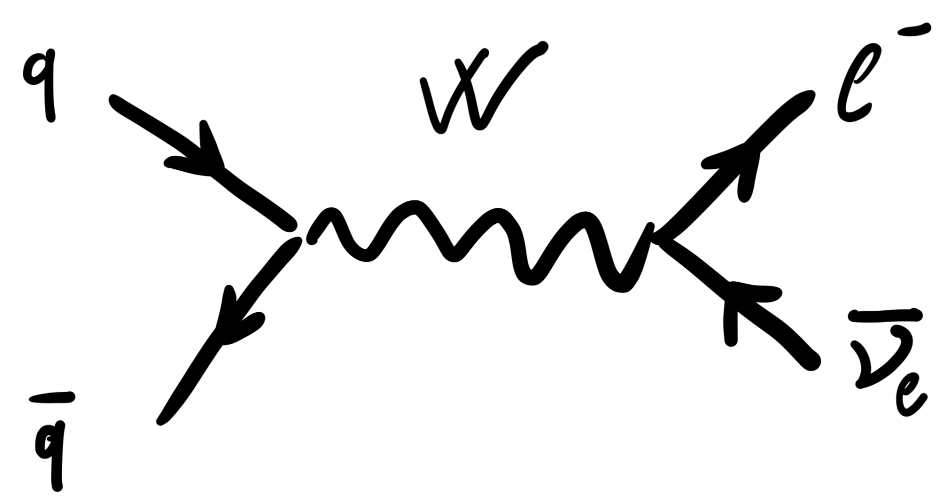


CHARGED CURRENT DRELL-YAN

*[T. Armadillo, R. Bonciani, **SD**, N.Rana, A.Vicini]*

-Work in progress-

- **56** Master integrals with two internal masses:
  - 34 **identical** to the ones appearing in the neutral current case;
  - 22 **generalisation** of masters with two massive lines to the case of different masses;
- Two possible approaches to compute the new masters:
  - Compute the boundary conditions with **AMFLOW** and use the **differential equations in *s* and *t*** to build the grid;
  - Use the grid of the neutral-current Drell-Yan as a boundary condition and use the **differential equation w.r.t. one of the masses**.
- **UV renormalisation** & **IR subtraction** are analogous to the neutral current case.



TOWARDS NNLO EW

FUTURE CHALLENGES

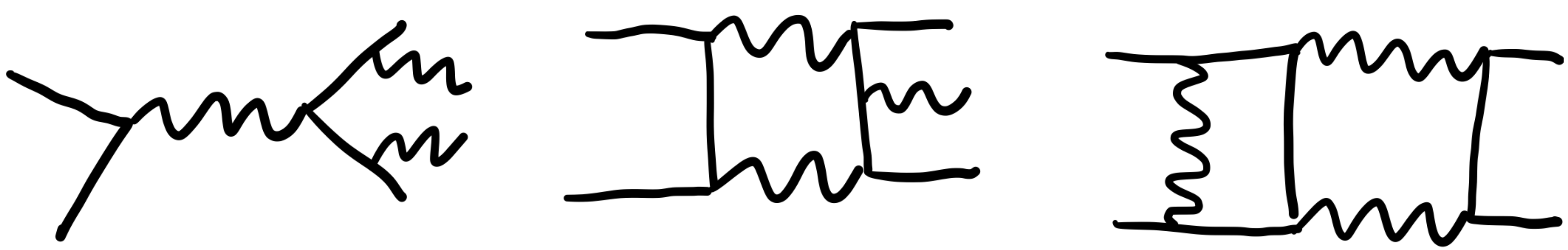
EXTRA CHALLENGES OF NNLO EW

What makes NNLO EW challenging?

- additional internal **massive lines**;
  additional scales in the problem (m_Z, m_W, m_H…)bring additional complications!

- treatment of γ_5;
  how can γ_5 be consistently used in dimensional regularisation?

- need for the **complex mass scheme**;
  requires to analytically continue the master integrals on the complex plane of the kinematical invariants!



POSSIBLE ISSUES OF NNLO EW

What makes NNLO EW challenging?

- additional internal **massive lines**;
  additional scales in the problem (m_Z, m_W, … *ions!*
- treatment of γ_5;
  how can γ_5 be consistently used in di…
- need for the **complex mass scheme**;
  requires to analytically continue the master integ… *kinematical invariants!*

Is the ABISS and KIRA running time going to be a problem?

Possibly complicated by the presence of fermionic triangles!

Is the SEASYDE running time going to be a problem?

In principle it is just the same more complicated, but technical difficulties possibly lie ahead!

TREATMENT OF γ_5

γ_5 is not well defined in a non integer number of dimensions!

	Anticommutation $\{\gamma_\mu, \gamma_5\} = 0$	Cyclicity of the trace
't Hooft and Veltmann *Nucl. Phys. B 44 (1972) 189–213*	✘	✔
Kreimer et al. *Phys. Lett. B 237 (1990) 59–62*	✔	✘

For neutral-current Drell Yan proven that at 2loops the two prescriptions yield:

➤ **different** scattering amplitudes; ➤ **same** finite corrections after subtraction.

[M. Heller, A. von Manteuffel, R. M. Schabinger and H. Spiesberger, arXiv:hep-ph/2012.05918]

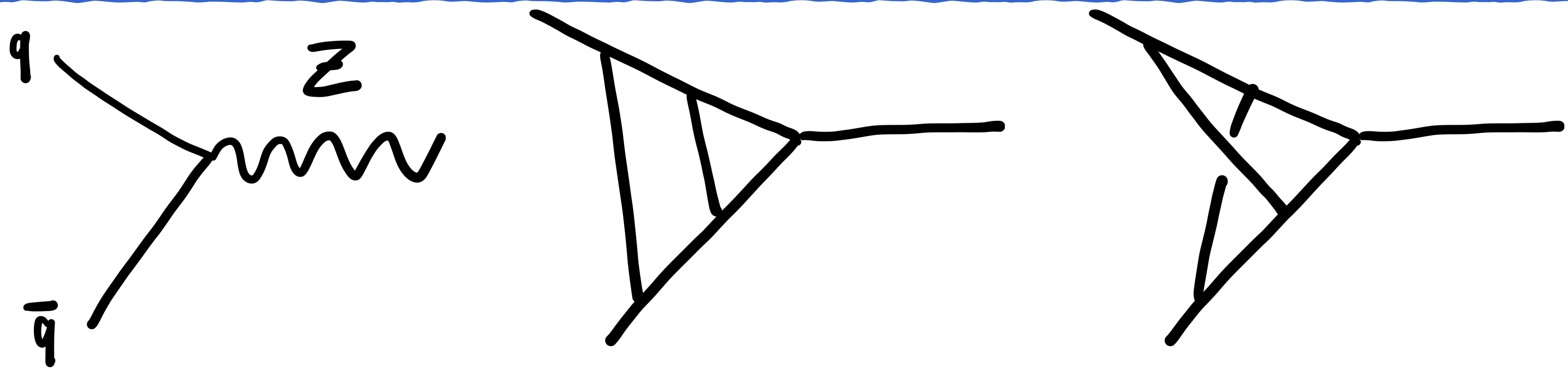
Our procedure for the mixed corrections:

1. Use anticommutation relation, bring all γ_5 at the end of the Dirac trace;
2. Use $\gamma_5^2 = 1$, end up with zero or one γ_5 in each Dirac trace;
3. Replace the (single) leftover γ_5 with the relation: $\gamma_5 = \frac{i}{4!}\epsilon_{\mu\nu\rho\sigma}\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma$.

How to handle now γ_5 in a generic NNLO EW computation?

TESTING THE WATERS: $q\bar{q}Z$

First test: 1particle irreducible contributions to $q\bar{q}Z$ @ NNLO EW



- Contributes to the **form factor** and the **amplitudes** for the process;
- **Proof of concept** for our framework;
- The computation involves:
  - **2236 diagrams**;
  - **52 integral families**;
  - **380 master integrals**;
  - up to **5 massive propagators** and **3 different masses**.

RUNNING TIMES

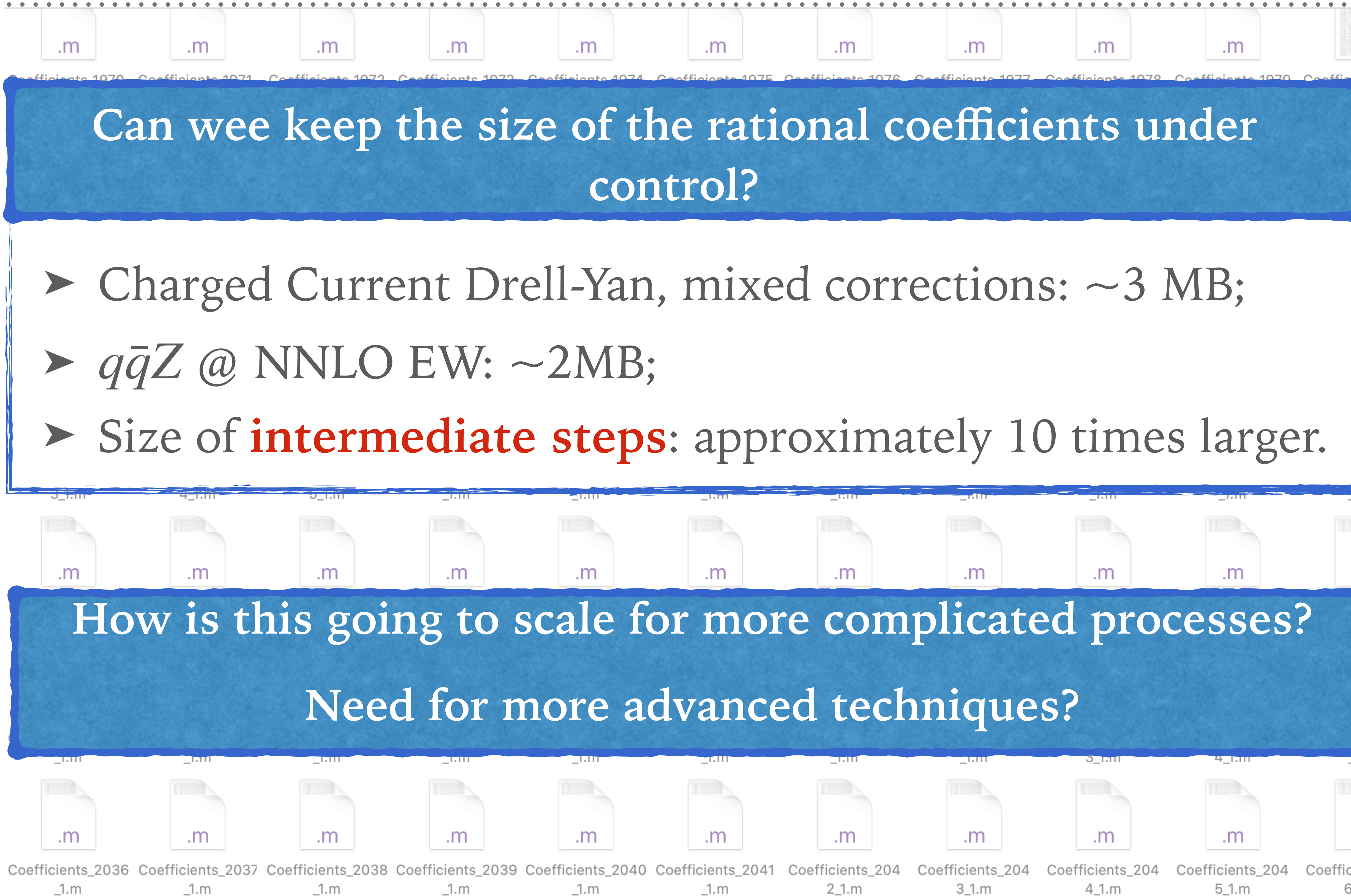
	NEUTRAL CURRENT DRELL-YAN	CHARGED CURRENT DRELL-YAN	$q\bar{q}Z$ NNLO EW
Most Complicated Integral Family	2 loop boxes with up to 2 massive lines and 1 different mass	2 loop boxes with up to 2 massive lines and 2 different masses	2 loop vertices with up to 5 massive lines and 3 different masses
Number of Masters	401	274	380
Run KIRA	12 h	16 h	1 d
Run AMFLOW (1 POINT)	1 d	1.5 d	2 d
Run SEASYDE (3250 POINTS)	$\sim$ 2 weeks (26 cores)	$\sim$ 3 weeks (26 cores)	//

SEASYDE runs are affected by our (limited) number of licenses!

RUNNING TIMES

SEASYDE	Run SeaSyde: 3250 points					
	Massive final state legs	# massive internal lines	# different masses	# masters	time	cores
Vertex 2L	✘	0	0	3	30 min	1
Vertex 2L	✘	1	1	6	2.5 h	1
Vertex 2L (non-planar)	✘	1	1	14	10 h	1
Box 2L	✘	2	1	36	5 d	26
Box 2L	✔	2	2	46	8 d	26
Box 2L	✘	2	2	56	10 d	26

RATIONAL COEFFICIENTS



Can wee keep the size of the rational coefficients under control?

- Charged Current Drell-Yan, mixed corrections: ~3 MB;
- $q\bar{q}Z$ @ NNLO EW: ~2MB;
- Size of **intermediate steps**: approximately 10 times larger.

How is this going to scale for more complicated processes?

Need for more advanced techniques?

SUMMARY & OUTLOOK

- **NNLO EW corrections** will be of crucial importance for the future LHC scientific program;
- We developed a **framework** which relies on ABISS for the evaluation of rational coefficients and SEASYDE for the evaluation of the Master Integrals;
- We successfully applied our framework to the computation of **mixed QCD-EW corrections** to the Drell-Yan process.

SUMMARY & OUTLOOK

- we want to use our framework to compute **NNLO EW** corrections;
- **proof of concept** for 1 particle irreducible contribution to $q\bar{q}Z$;
- **Challenges** are still ahead:
  - increased number of scales;
  - treatment of γ_5;
  - …
- **Are more refined strategies required?**

SUMMARY & OUTLOOK

- we want to use our framework to compute **NNLO EW** corrections;
- **proof of concept** for 1 particle irreducible contribution to $q\bar{q}Z$;
- **Challenges** are still ahead:
  - increased number of scales;
  - treatment of γ_5;
  - …
- **Are more refined strategies required?**

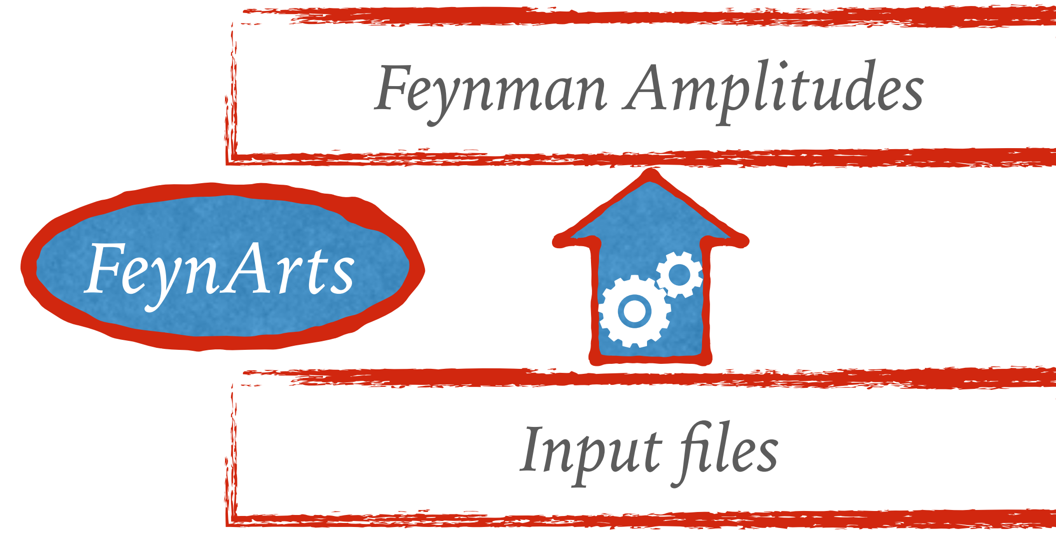
THANKS!

BACKUP SLIDES

GENERATION OF THE AMPLITUDES

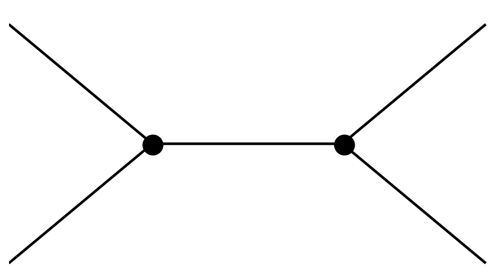
FeynArts - *arXiv:hep-ph/0012260*

Mathematica package for the generation and visualisation of Feynman diagrams and amplitudes. Several models are available.

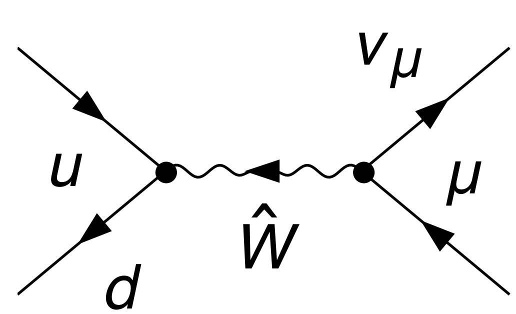


- FeynArts work-flow:

*Creation of the **topologies*** → *Insertion of **fields** into the topologies* → *Application of the Feynman rules to produce **Feynman amplitudes***



T1



T1 P1 N1

```
{v̄[p2, 0].
   (i EL gWdu CKM[1, 1]* ga[Lor1].om_).
   u[p1, 0] ū[k1, 0].
   (i EL gWNl ga[Lor2].om_).v[k2, 0]
 g[Lor1, Lor2]      1
               -----------------}
               -MW² + (k1 + k2)²
```

COMPUTATION OF THE INTERFERENCE TERMS

In **ABISS** are included routines to automatise the computation of interferences, in particular:

- **Lorentz algebra** (handle scalar products)
- **Dirac algebra** (compute traces of gamma matrices)

Tensor Integrals

Feynman Amplitudes

- The result can be written as a sum of **tensor integrals** in the form:

$$\int \prod_{i=1}^{L} dq_i \, \frac{q_1^{\mu_1}\ldots q_1^{\mu_j}\ldots q_L^{\mu_1}\ldots q_L^{\mu_l}}{\mathscr{P}_1\ldots\mathscr{P}_t}$$

where:

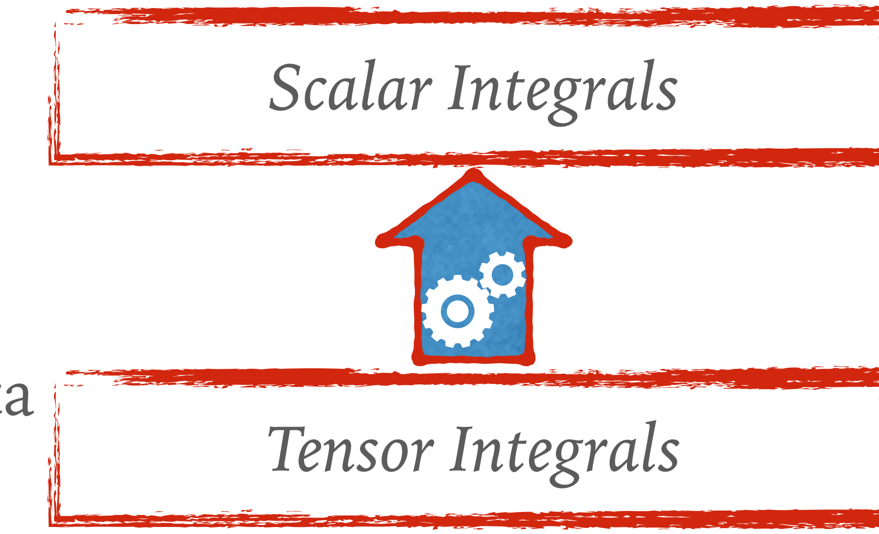
- $q_i \rightarrow$ loop momentum;
- $L \rightarrow$ number of independent loop momenta;
- $\mathscr{P}_i = k_i^2 - m^2 \rightarrow$ inverse propagator, k_i linear combination of momenta.

<u>ISSUE</u>: **Handling γ_5 in dimensional regularisation.**

Object inherently 4D: how can we use it in arbitrary space-time dimension?

SCALAR INTEGRALS

- We rearrange the expressions in terms of **scalar integrals**, better suited for the **reduction algorithms**;



- Write all the scalar products involving loop momenta in terms of **inverse propagators**:

$$\begin{cases} \mathscr{P}_0 = q^2 - m_0^2 \\ \mathscr{P}_1 = (q-p_1)^2 - m_1^2 \end{cases} \Rightarrow \begin{cases} q^2 = \mathscr{P}_0 + m_0^2 \\ q \cdot p_1 = 1/2\,(\mathscr{P}_0 + m_0 - \mathscr{P}_1 - m_1) \end{cases}$$

- To this end, it is necessary to introduce **auxiliary propagators** $\mathscr{P}_{t+1}^{\alpha_t+1} \ldots \mathscr{P}_N^{\alpha_N}$ to close the algebra.

Tensor Integral

$$\int \prod_{i=1}^{L} d^n q_i \, \frac{q_1^{\mu_1} \ldots q_1^{\mu_j} \ldots q_L^{\mu_1} \ldots q_L^{\mu_l}}{\mathscr{P}_1 \ldots \mathscr{P}_t}$$

$\rightarrow$

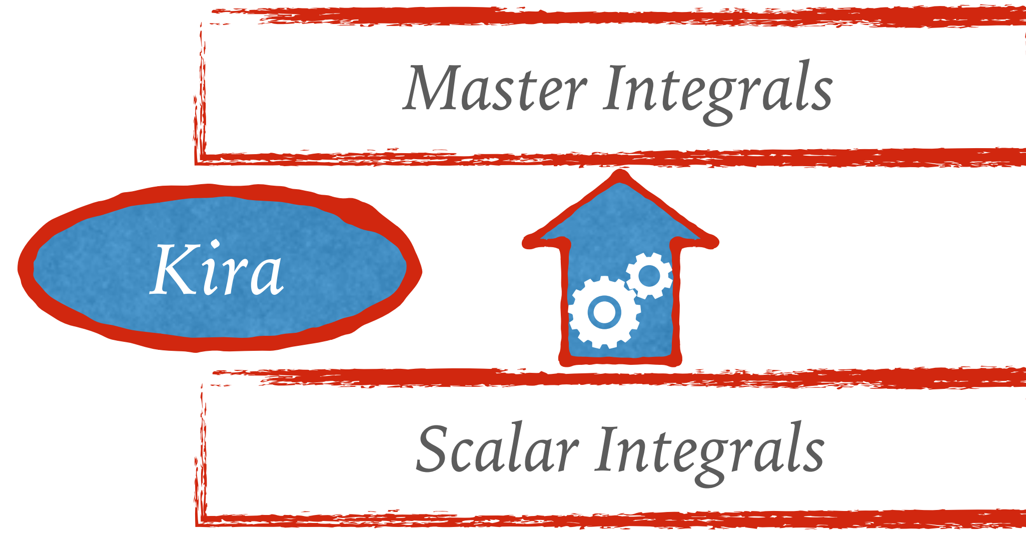
Scalar Integral

$$\int \prod_{i=1}^{L} d^n q_i \, \frac{1}{\mathscr{P}_1^{\alpha_1} \ldots \mathscr{P}_t^{\alpha_t} \mathscr{P}_{t+1}^{\alpha_{t+1}} \ldots \mathscr{P}_N^{\alpha_N}}$$

REDUCTION TO MASTER INTEGRALS

Kira - *arXiv:hep-ph/1705.05610*
- *arXiv:hep-ph/2008.06494*

C++ reduction program implementing Laporta algorithm



Scalar Integrals $\sum_{i=1}^{N} \hat{c}_i I_{S,i}$ → **Master Integrals** $\sum_{i=1}^{n} c_i I_{M,i} \qquad n \ll N$

- Expressions written as a sum of **scalar integrals** with the respective coefficient;
- All the scalar integrals are not independent: linear relations between them are provided by **integration by parts (IBP) identities**;
- We can reduce the large set of scalar integrals to a smaller set of **master integrals**;
- Kira applies **Laporta algorithm** to apply IBP identities to a set of integrals in order to find the linear relations between them.

INTEGRATION BY PARTS IDENTITIES

Gauss Theorem: $\int d^n q \frac{\partial}{\partial q^\mu} f^\mu(p_i^\mu, \cdots, q_i^\mu) = 0$

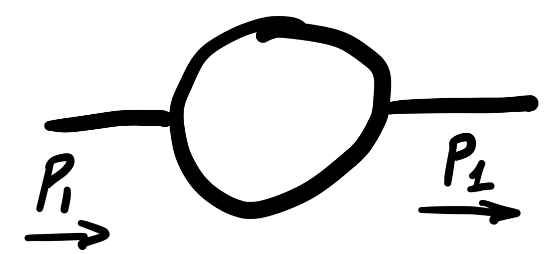
We choose, e.g.

$$f^\mu(p_i^\mu, \cdots, q_i^\mu) = \frac{p_1^\mu}{(q^2 - m_0^2)^{\alpha_0}((q+p_1)^2 - m_1^2)^{\alpha_1}}$$

$$\Downarrow$$

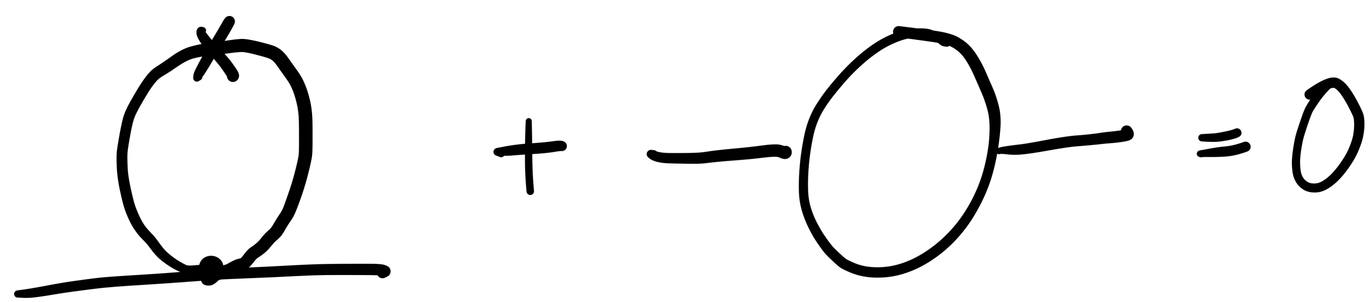
$$\int d^n q \frac{\partial}{\partial q^\mu} f^\mu = - \int d^n q \left(\frac{2\alpha_0\, q \cdot p_1}{(q^2 - m_0^2)^{\alpha_0+1}((q+p_1)^2 - m_1^2)^{\alpha_1}} + \frac{2\alpha_1\,(q+p_1)\cdot p_1}{(q^2 - m_0^2)^{\alpha_0}((q+p_1)^2 - m_1^2)^{\alpha_1+1}} \right) = 0$$

Example: $\alpha_0 = 1, \quad \alpha_1 = 1, \quad p_1^2 = 0, \quad m_0 = m_1 = m.$



$$\begin{cases} \mathscr{P}_0 = q^2 - m^2 \\ \mathscr{P}_1 = (q+p_1)^2 - m^2 \end{cases}$$

$$\begin{cases} q^2 = \mathscr{P}_0 + m^2 \\ q \cdot p_1 = \frac{1}{2}(\mathscr{P}_0 + \mathscr{P}_1) \end{cases}$$



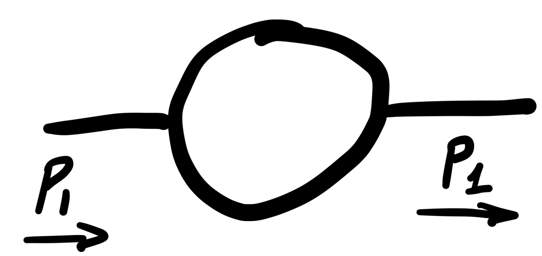
By using different IBP relations it is possible to define **ladder operators** to rise and lower the indices of the powers of the propagators!

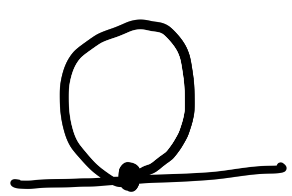
MASTERS AS SOLUTIONS OF DIFFERENTIAL EQUATIONS

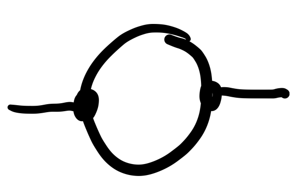
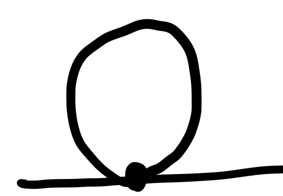
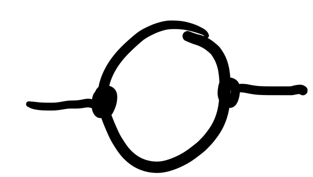
TO DO: Solve the Master Integrals. → **Method of differential equations**

- By **deriving with respect to one of the kinematic variables** one of the master integrals, we obtain a new scalar integral;
- By using the **IBP identities**, any scalar integral can be written in terms of Master Integrals;
- We can obtain a **system of differential equations for the Master Integrals**!

Example

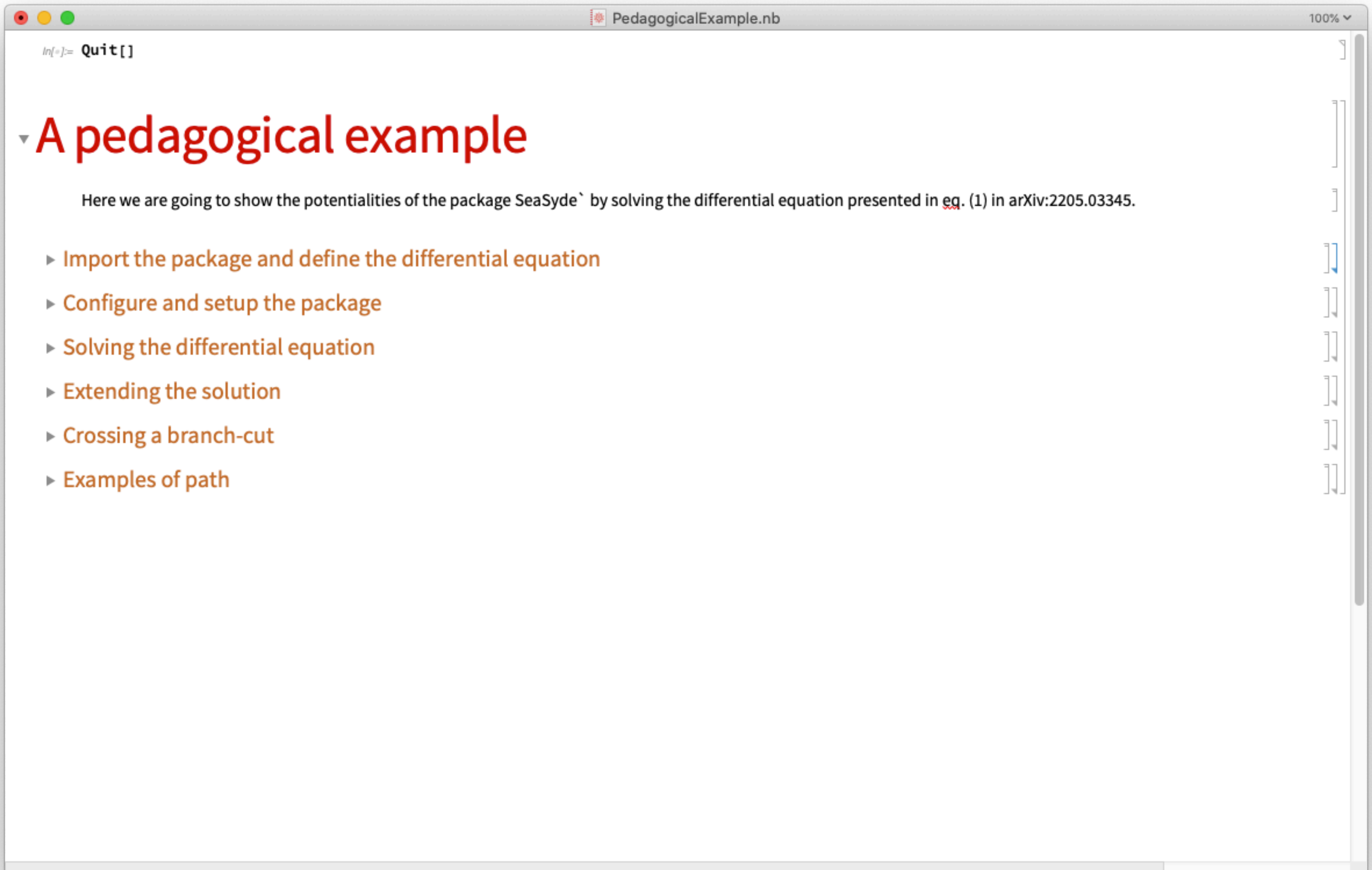


$$\frac{d}{dp_1^2}$$  $$= 0$$

$$\frac{d}{dp_1^2}$$  $$= \frac{d-2}{p_1^2(4m^2-p_1^2)}$$  $$- \frac{(d-4)p_1^2+4m^2}{2p_1^2(4m^2-p_1^2)}$$ 

SEASYDE

NOTEBOOK EXAMPLE



SEASYDE

NOTEBOOK EXAMPLE

PedagogicalExample.nb

```
In[•]:= Quit[]
```

A pedagogical example

Here we are going to show the potentialities of the package SeaSyde` by solving the differential equation presented in eq. (1) in arXiv:2205.03345.

Import the package and define the differential equation

```
SetDirectory[NotebookDirectory[]];
<< ../SeaSyde.m
```

$$\text{In[•]:= Equation} = \left\{f'[x] + \frac{1}{x^2 - 4x + 5} f[x] == \frac{1}{x+2}\right\};$$

```
BoundaryCondition = {f[0] == 1};

MasterIntegral = {f[x]};
PointBC = {0};
```

Configure and setup the package

Solving the differential equation

Extending the solution

Crossing a branch-cut

Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

```
In[•]:= Quit[]
```

A pedagogical example

Here we are going to show the potentialities of the package SeaSyde` by solving the differential equation presented in eq. (1) in arXiv:2205.03345.

Import the package and define the differential equation

```
In[1]:= SetDirectory[NotebookDirectory[]];
<< ../SeaSyde.m
```

```
+++++ SeaSyde` +++++
Version 1.1.0
SeaSyde is a package for solving the system of differential equation associated to the Master Integrals of a given topology.
For any question or comment, please contact:
T. Armadillo, R. Bonciani, S. Devoto, N. Rana or A. Vicini.
For the latest version please see the GitHub repository.
```

In[3]:= Equation = $\left\{f'[x] + \frac{1}{x^2 - 4x + 5} f[x] == \frac{1}{x+2}\right\}$;

```
BoundaryCondition = {f[0] == 1};

MasterIntegral = {f[x]};
PointBC = {0};
```

Configure and setup the package

Solving the differential equation

Extending the solution

Crossing a branch-cut

NOTEBOOK EXAMPLE

PedagogicalExample.nb

```
In[•]:= Quit[]
```

A pedagogical example

Here we are going to show the potentialities of the package SeaSyde` by solving the differential equation presented in eq. (1) in arXiv:2205.03345.

Import the package and define the differential equation

Configure and setup the package

```
In[•]:= Configuration = {
          EpsilonOrder → 0,
          ExpansionOrder → 50
        };
        UpdateConfiguration[Configuration]
        SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]
```

Solving the differential equation

Extending the solution

Crossing a branch-cut

Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

```
In[•]:= Quit[]
```

A pedagogical example

Here we are going to show the potentialities of the package SeaSyde` by solving the differential equation presented in eq. (1) in arXiv:2205.03345.

Import the package and define the differential equation

Configure and setup the package

```
In[7]:= Configuration = {
          EpsilonOrder → 0,
          ExpansionOrder → 50
        };
        UpdateConfiguration[Configuration]
        SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]

SeaSyde: Updated EpsilonOrder parameter, new value -> 0
SeaSyde: Updated ExpansionOrder parameter, new value -> 50
SeaSyde: There are 1 kinematics variables {x}
SeaSyde: The Feynman prescriptions for the variables are {i δ}
SeaSyde: There are 1 Master Integrals
SeaSyde: The boundary conditions are imposed in x == 0.
SeaSyde: The boundary conditions are given as precise value of the solution.
SeaSyde: There are 1 equations and 1 boundary conditions
SeaSyde: The possible singularities for the kinematics variables {x} are respectively {{-2., 2. - 1. i, 2. + 1. i}}
SeaSyde: The system of differential equation has been set and expanded in ϵ
```

Solving the differential equation

Extending the solution

NOTEBOOK EXAMPLE

PedagogicalExample.nb

Solving the differential equation

We can solve the equation in the point where the boundary conditions are imposed, e.g. x=0.

In[•]:= `Equation`

Out[•]= $\left\{\frac{f[x]}{5-4x+x^2}+f'[x]=\frac{1}{2+x}\right\}$

In[•]:= `SolveSystem[x]`

We can compare the first few terms with the exact result provided in eq. (3), (4) and (5).

In[•]:= $\text{ExactResult}=\frac{1}{2}x-\frac{7}{40}x^2+\frac{2}{75}x^3+\frac{1}{5}\left(5-x-\frac{3}{10}x^2-\frac{11}{150}x^3\right)$ `// Expand // N`

In[•]:= `Solution[] /. x^b_ /; b > 3 -> 0 // N`

The singularities are in x=-2, x=2±*i*. The closest to the centre of the series, x=0, is x=-2, hence the radius of convergence is ρ=|-2-0|=2. We can see that explicitly by plotting the solution along the real axes.

In[•]:= `Plot[Solution[], {x, -2.3, 2.3}]`

If we go over x=2 the solution does not converge anymore.

Extending the solution

Crossing a branch-cut

Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

Solving the differential equation

We can solve the equation in the point where the boundary conditions are imposed, e.g. x=0.

```
In[10]:= Equation
```

Out[10]= $\left\{\frac{f[x]}{5-4x+x^2}+f'[x]==\frac{1}{2+x}\right\}$

```
In[11]:= SolveSystem[x]
```

```
Solving equation for ϵ order 0
```

I solved the system of equation. The error estimate is: 5.07431×10^{-17}.

We can compare the first few terms with the exact result provided in eq. (3), (4) and (5).

In[•]:= $\text{ExactResult} = \frac{1}{2}x - \frac{7}{40}x^2 + \frac{2}{75}x^3 + \frac{1}{5}\left(5 - x - \frac{3}{10}x^2 - \frac{11}{150}x^3\right)$ // Expand // N

In[•]:= Solution[] /. $x^{b\_}$ /; $b > 3$ -> 0 // N

The singularities are in x=-2, x=2±*i*. The closest to the centre of the series, x=0, is x=-2, hence the radius of convergence is ρ=|-2-0|=2. We can see that explicitly by plotting the solution along the real axes.

```
In[•]:= Plot[Solution[], {x, -2.3, 2.3}]
```

If we go over x=2 the solution does not converge anymore.

Extending the solution

Crossing a branch-cut

Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

Solving the differential equation

We can solve the equation in the point where the boundary conditions are imposed, e.g. x=0.

In[10]:= `Equation`

Out[10]= $\left\{\frac{f[x]}{5-4x+x^2}+f'[x]==\frac{1}{2+x}\right\}$

In[11]:= `SolveSystem[x]`

```
Solving equation for ϵ order 0
I solved the system of equation. The error estimate is: 5.07431×10^-17.
```

We can compare the first few terms with the exact result provided in eq. (3), (4) and (5).

In[12]:= `ExactResult =` $\frac{1}{2}x-\frac{7}{40}x^2+\frac{2}{75}x^3+\frac{1}{5}\left(5-x-\frac{3}{10}x^2-\frac{11}{150}x^3\right)$ `// Expand // N`

Out[12]= $1.+0.3x-0.235x^2+0.012x^3$

`Solution[] /. x^b_ /; b > 3 -> 0 // N`

The singularities are in x=-2, x=2±*i*. The closest to the centre of the series, x=0, is x=-2, hence the radius of convergence is ρ=|-2-0|=2. We can see that explicitly by plotting the solution along the real axes.

In[ ]:= `Plot[Solution[], {x, -2.3, 2.3}]`

If we go over x=2 the solution does not converge anymore.

Extending the solution

Crossing a branch-cut

Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

Solving the differential equation

We can solve the equation in the point where the boundary conditions are imposed, e.g. x=0.

In[10]:= `Equation`

Out[10]= $\left\{\frac{f[x]}{5-4x+x^2}+f'[x]=\frac{1}{2+x}\right\}$

In[11]:= `SolveSystem[x]`

```
Solving equation for ϵ order 0
```

I solved the system of equation. The error estimate is: 5.07431×10^{-17}.

We can compare the first few terms with the exact result provided in eq. (3), (4) and (5).

In[12]:= ExactResult = $\frac{1}{2}x-\frac{7}{40}x^2+\frac{2}{75}x^3+\frac{1}{5}\left(5-x-\frac{3}{10}x^2-\frac{11}{150}x^3\right)$ // Expand // N

Out[12]= $1.+0.3x-0.235x^2+0.012x^3$

In[14]:= Solution[] /. $x^{b\_}$ /; $b>3$ -> 0 // N

Out[14]= $\{1.+0.3x-0.235x^2+0.012x^3\}$

The singularities are in x=-2, x=2±*i*. The closest to the centre of the series, x=0, is x=-2, hence the radius of convergence is ρ=|-2-0|=2. We can see that explicitly by plotting the solution along the real axes.

In[ ]:= `Plot[Solution[], {x, -2.3, 2.3}]`

If we go over x=2 the solution does not converge anymore.

Extending the solution

Crossing a branch-cut

Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

In[10]:= Equation

Out[10]= $\left\{\frac{f[x]}{5-4x+x^2}+f'[x]==\frac{1}{2+x}\right\}$

```
In[11]:= SolveSystem[x]
Solving equation for ϵ order 0
```

I solved the system of equation. The error estimate is: 5.07431×10^{-17}.

We can compare the first few terms with the exact result provided in eq. (3), (4) and (5).

In[12]:= ExactResult = $\frac{1}{2}x-\frac{7}{40}x^2+\frac{2}{75}x^3+\frac{1}{5}\left(5-x-\frac{3}{10}x^2-\frac{11}{150}x^3\right)$ // Expand // N

Out[12]= $1.+0.3x-0.235x^2+0.012x^3$

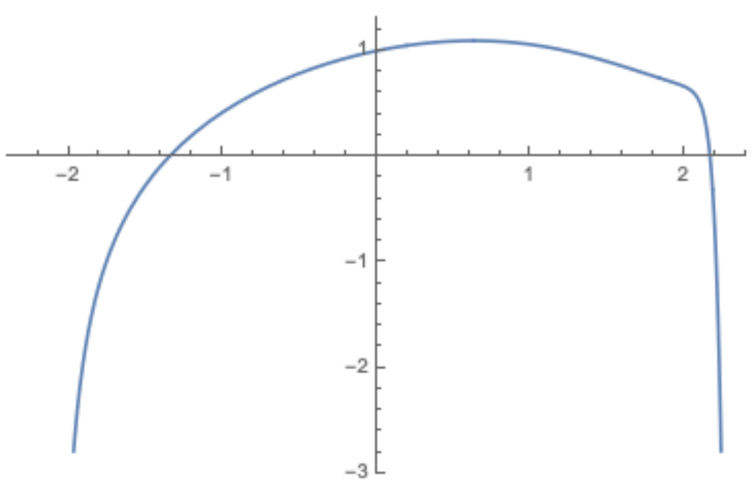
In[14]:= Solution[] /. $x^{b\_}$ /; $b>3$ -> 0 // N

Out[14]= $\{1.+0.3x-0.235x^2+0.012x^3\}$

The singularities are in x=-2, x=2±*i*. The closest to the centre of the series, x=0, is x=-2, hence the radius of convergence is ρ=|-2-0|=2. We can see that explicitly by plotting the solution along the real axes.

```
In[15]:= Plot[Solution[], {x, -2.3, 2.3}]
```

Out[15]=



If we go over x=2 the solution does not converge anymore.

NOTEBOOK EXAMPLE

PedagogicalExample.nb

- Solving the differential equation
- Extending the solution

If we want the solution in another point, e.g. x=-1+0.5*i*, we can attach multiple series solutions.

In[•]:=
```
TransportBoundaryConditions[{-1 + 0.5 i}]
```

And we can access the value of the solution in x=-1+0.5*i* by using the SolutionValue[] method.

In[•]:=
```
SolutionValue[] // N
```

- Crossing a branch-cut
- Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

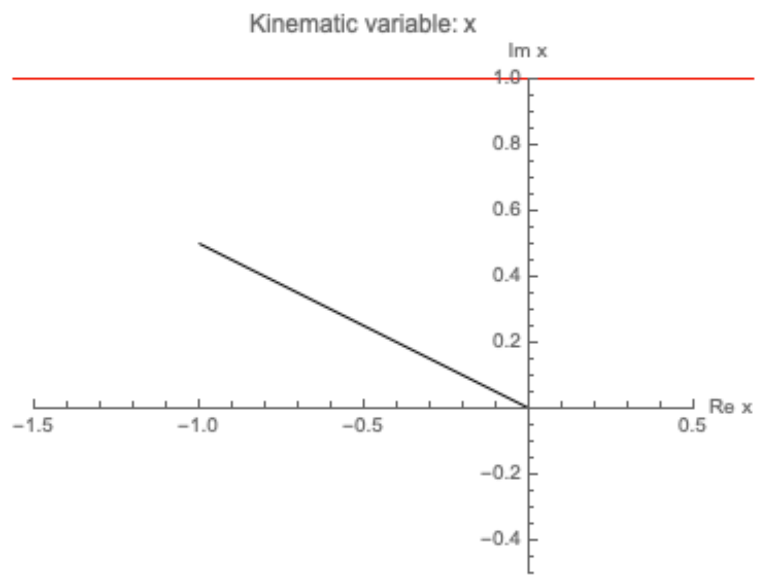
Solving the differential equation

Extending the solution

If we want the solution in another point, e.g. x=-1+0.5*i*, we can attach multiple series solutions.

```
In[16]:= TransportBoundaryConditions[{-1 + 0.5 ⅈ}]
```

```
SeaSyde: Moving following these points: {0., -1. + 0.5 ⅈ}
, avoiding singularities. Here you can see the path in the complex plane for the kinematic variable x
```




```
SeaSyde: Moving from the point x=0. to x=-1. + 0.5 ⅈ, along the line x=(-1. + 0.5 ⅈ) tInt
SeaSyde: The new point is: x=-0.894427 + 0.447214 ⅈ
SeaSyde: I arrived at x=-1. + 0.5 ⅈ. The error estimate is: 5.07431×10^-17.
Use Solution[] or SolutionValue[] to access the solution, or CreteGraph to plot it.
```

And we can access the value of the solution in x=-1+0.5*i* by using the SolutionValue[] method.

```
In[ ]:= SolutionValue[] // N
```

Crossing a branch-cut

Examples of path

SEASYDE

NOTEBOOK EXAMPLE

PedagogicalExample.nb

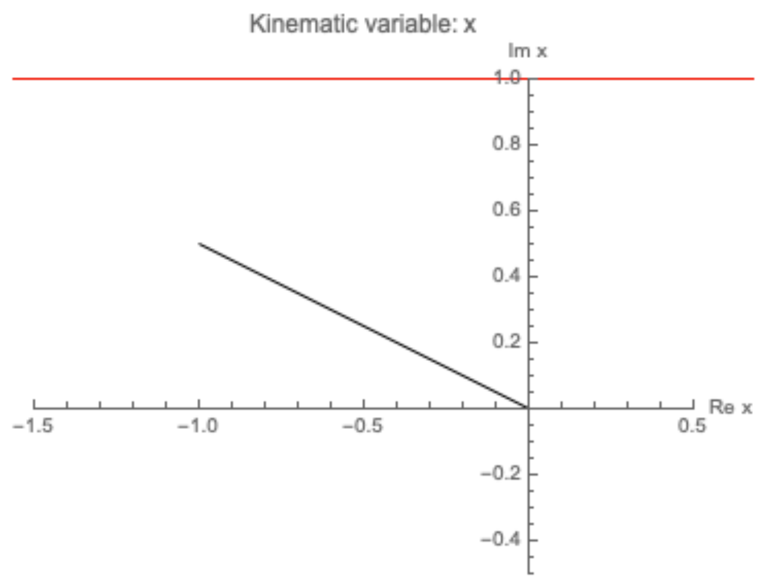
100%

Solving the differential equation

Extending the solution

If we want the solution in another point, e.g. x=-1+0.5*i*, we can attach multiple series solutions.

```
In[16]:= TransportBoundaryConditions[{-1 + 0.5 i}]

SeaSyde: Moving following these points: {0., -1. + 0.5 i}
 , avoiding singularities. Here you can see the path in the complex plane for the kinematic variable x
```




```
SeaSyde: Moving from the point x=0. to x=-1. + 0.5 i, along the line x=(-1. + 0.5 i) tInt
SeaSyde: The new point is: x=-0.894427 + 0.447214 i
SeaSyde: I arrived at x=-1. + 0.5 i. The error estimate is: 5.07431×10^-17.
Use Solution[] or SolutionValue[] to access the solution, or CreteGraph to plot it.
```

And we can access the value of the solution in x=-1+0.5*i* by using the SolutionValue[] method.

```
In[17]:= SolutionValue[] // N
Out[17]= {0.545912 + 0.443234 i}
```

Crossing a branch-cut

NOTEBOOK EXAMPLE

PedagogicalExample.nb

▸ Solving the differential equation

▸ Extending the solution

▾ Crossing a branch-cut

We might want to overcome a branch-cut. In order to do so it is important to circumvent the singularities to the right

```
In[•]:= SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]
        TransportVariable[x, 2 i]
```

```
In[•]:= SolutionValue[] // N
```

If we cross the branch-cut directly, the result might be different

```
In[•]:= SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]
        TransportVariable[x, 2 i, CreateLine[{0, 2 i}]]
```

```
In[•]:= SolutionValue[] // N
```

We observe that the solution is different from the previous case. This is because by crossing the cut directly, we end up on another Riemann sheet.

▸ Examples of path

NOTEBOOK EXAMPLE

PedagogicalExample.nb

In[18]:=
```
SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]
TransportVariable[x, 2 i]
```

SeaSyde: There are 1 kinematics variables {x}

SeaSyde: The Feynman prescriptions for the variables are {i δ}

SeaSyde: There are 1 Master Integrals

SeaSyde: The boundary conditions are imposed in x == 0.

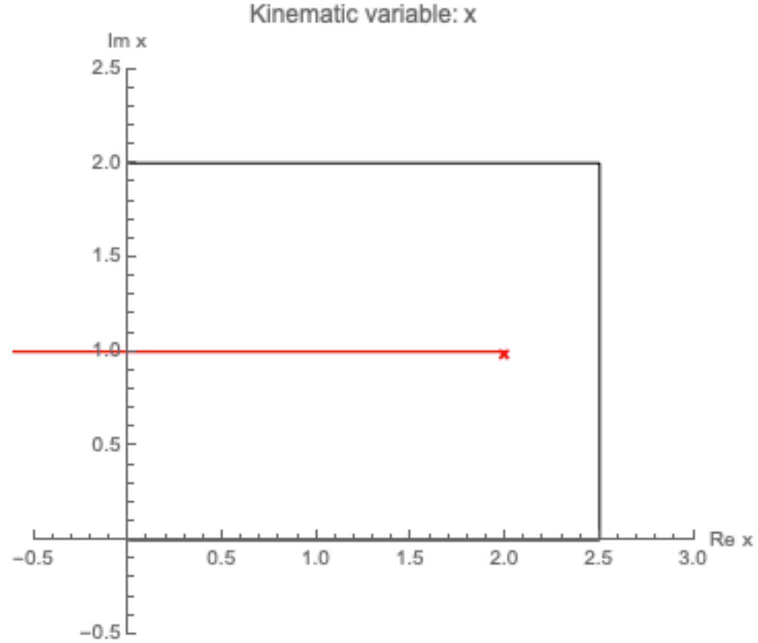
SeaSyde: The boundary conditions are given as precise value of the solution.

SeaSyde: There are 1 equations and 1 boundary conditions

SeaSyde: The possible singularities for the kinematics variables {x} are respectively {{-2., 2. - 1. i, 2. + 1. i}}

SeaSyde: The system of differential equation has been set and expanded in ϵ

SeaSyde: Moving following these points: {0., 2.5, 2.5 + 2. i, 0. + 2. i}
, avoiding singularities. Here you can see the path in the complex plane for the kinematic variable x



SeaSyde: Moving from the point x=0. to x=2.5, along the line x=2.5 tInt

SeaSyde: The new point is: x=1.

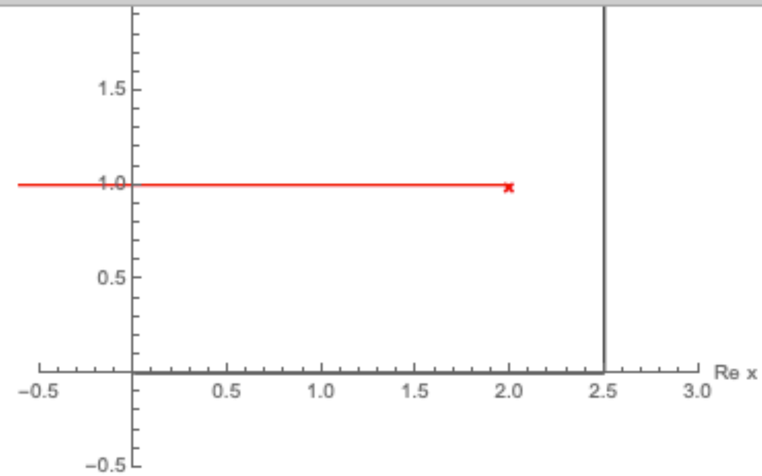
SeaSyde: The new point is: x=1.70711

SeaSyde: The new point is: x=2.22811

SeaSyde: Moving from the point x=2.5 to x=2.5 + 2. i, along the line x=2.5 + (0. + 2. i) tInt

NOTEBOOK EXAMPLE

PedagogicalExample.nb

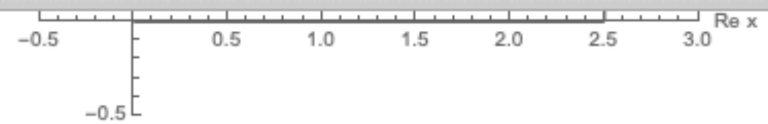



```
SeaSyde: Moving from the point x=0. to x=2.5, along the line x=2.5 tInt
SeaSyde: The new point is: x=1.
SeaSyde: The new point is: x=1.70711
SeaSyde: The new point is: x=2.22811
SeaSyde: Moving from the point x=2.5 to x=2.5 + 2. i, along the line x=2.5 + (0. + 2. i) tInt
SeaSyde: The new point is: x=2.5 + 0.559017 i
SeaSyde: The new point is: x=2.5 + 0.892358 i
SeaSyde: The new point is: x=2.5 + 1.14809 i
SeaSyde: The new point is: x=2.5 + 1.40882 i
SeaSyde: The new point is: x=2.5 + 1.73175 i
SeaSyde: Moving from the point x=2.5 + 2. i to x=0. + 2. i, along the line x=(2.5 + 2. i) - 2.5 tInt
SeaSyde: The new point is: x=1.94098 + 2. i
SeaSyde: The new point is: x=1.44011 + 2. i
SeaSyde: The new point is: x=0.867079 + 2. i
SeaSyde: The new point is: x=0.111514 + 2. i
SeaSyde: I arrived at x=0. + 2. i. The error estimate is: 5.29003×10^-18.
Use Solution[] or SolutionValue[] to access the solution, or CreteGraph to plot it.
```

```
In[20]:= SolutionValue[] // N
Out[20]= {-0.424527 + 0.399169 i}
```

NOTEBOOK EXAMPLE

PedagogicalExample.nb




```
SeaSyde: Moving from the point x=0. to x=2.5, along the line x=2.5 tInt
SeaSyde: The new point is: x=1.
SeaSyde: The new point is: x=1.70711
SeaSyde: The new point is: x=2.22811
SeaSyde: Moving from the point x=2.5 to x=2.5 + 2. i, along the line x=2.5 + (0. + 2. i) tInt
SeaSyde: The new point is: x=2.5 + 0.559017 i
SeaSyde: The new point is: x=2.5 + 0.892358 i
SeaSyde: The new point is: x=2.5 + 1.14809 i
SeaSyde: The new point is: x=2.5 + 1.40882 i
SeaSyde: The new point is: x=2.5 + 1.73175 i
SeaSyde: Moving from the point x=2.5 + 2. i to x=0. + 2. i, along the line x=(2.5 + 2. i) - 2.5 tInt
SeaSyde: The new point is: x=1.94098 + 2. i
SeaSyde: The new point is: x=1.44011 + 2. i
SeaSyde: The new point is: x=0.867079 + 2. i
SeaSyde: The new point is: x=0.111514 + 2. i
SeaSyde: I arrived at x=0. + 2. i. The error estimate is: 5.29003×10^-18.
Use Solution[] or SolutionValue[] to access the solution, or CreteGraph to plot it.

In[20]:= SolutionValue[] // N

Out[20]= {0.545912 + 0.443234 i}
```

If we cross the branch-cut directly, the result might be different

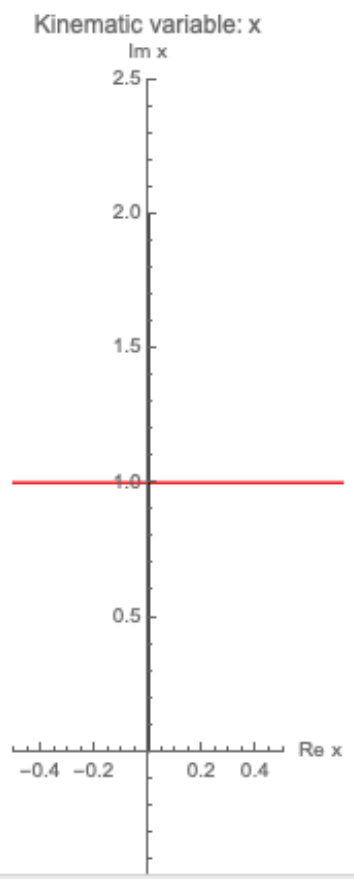
```
In[ ]:= SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]
TransportVariable[x, 2 i, CreateLine[{0, 2 i}]]

In[ ]:= SolutionValue[] // N

Out[ ]= {1.66537 + 0.582592 i}
```

We observe that the solution is different from the previous case. This is because by crossing the cut directly, we end up on another Riemann sheet.

NOTEBOOK EXAMPLE

PedagogicalExample.nb

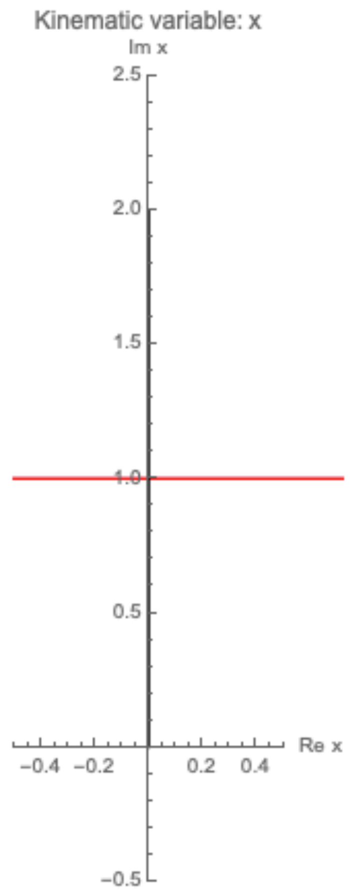
```
In[21]:= SetSystemOfDifferentialEquation[Equation, BoundaryCondition, MasterIntegral, {x + I δ}, PointBC]
TransportVariable[x, 2 i, CreateLine[{0, 2 i}]]

SeaSyde: There are 1 kinematics variables {x}
SeaSyde: The Feynman prescriptions for the variables are {i δ}
SeaSyde: There are 1 Master Integrals
SeaSyde: The boundary conditions are imposed in x == 0.
SeaSyde: The boundary conditions are given as precise value of the solution.
SeaSyde: There are 1 equations and 1 boundary conditions
SeaSyde: The possible singularities for the kinematics variables {x} are respectively {{-2., 2. - 1. i, 2. + 1. i}}
SeaSyde: The system of differential equation has been set and expanded in ϵ
SeaSyde: Moving following these points: {0., 0. + 2. i}
 , avoiding singularities. Here you can see the path in the complex plane for the kinematic variable x
```



NOTEBOOK EXAMPLE

PedagogicalExample.nb

SeaSyde: Moving following these points: {0., 0. + 2. i}
, avoiding singularities. Here you can see the path in the complex plane for the kinematic variable x



SeaSyde: Moving from the point x=0. to x=0. + 2. i, along the line x=(0. + 2. i) tInt

SeaSyde: The new point is: x=0. + 1. i

SeaSyde: The new point is: x=0. + 2. i

SeaSyde: I arrived at x=0. + 2. i. The error estimate is: 1.2243×10^{-16}.
Use Solution[] or SolutionValue[] to access the solution, or CreteGraph to plot it.

In[23]:= **SolutionValue[] // N**

Out[23]= {1.66537 + 0.582592 i}

We observe that the solution is different from the previous case. This is because by crossing the cut directly, we end up on another Riemann sheet.