UNIVERSITY OF LIVERPOOL

LEVERHULME TRUST

Fast evaluation of Feynman integrals for MC generators

Pau Petit Rosàs

In collaboration with W. Torres Bobadilla

LIVERPOOL THEORETICAL PHYSICS

Motivation

Amplitude

Master Integrals (MIs)

Evaluation of MIs

Motivation

Qgraf, Recola, FeynRules, FORM, FeynCalc, Tapir, …

Amplitude → Master Integrals (MIs) → Evaluation of MIs

Motivation

Qgraf, Recola, FeynRules, FORM, FeynCalc, Tapir, ...

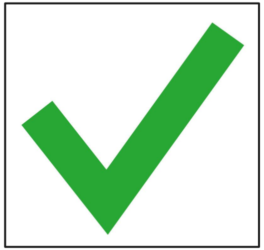
Amplitude → Master Integrals (MIs) → Evaluation of MIs

Motivation

Qgraf, Recola, FeynRules, FORM, FeynCalc, Tapir, …

Kira, Fire, Blade, LiteRed, NeatIBP, Reduze,...

Amplitude → Master Integrals (MIs) → Evaluation of MIs

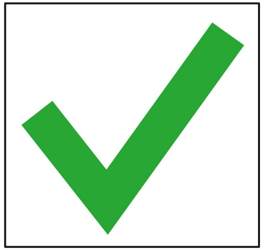


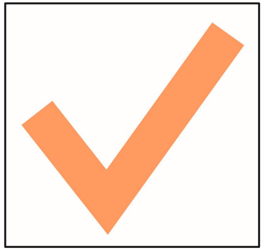
Motivation

Qgraf, Recola, FeynRules, FORM, FeynCalc, Tapir, …

Kira, Fire, Blade, LiteRed, NeatIBP, Reduze,...

Amplitude → Master Integrals (MIs) → Evaluation of MIs





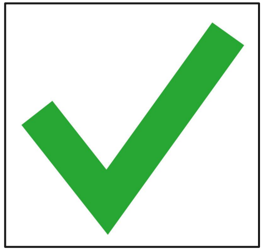
Motivation

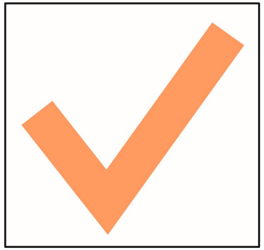
Qgraf, Recola, FeynRules, FORM, FeynCalc, Tapir, …

Kira, Fire, Blade, LiteRed, NeatIBP, Reduze,...

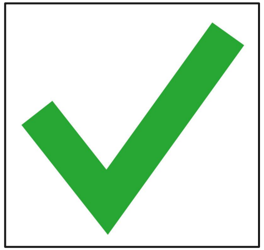
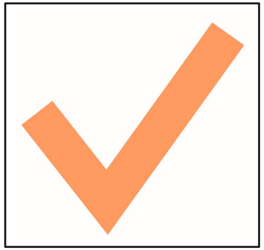
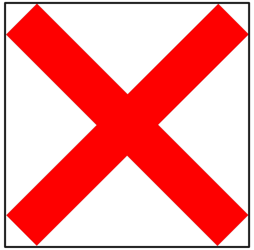
AMFlow, pySecDec, DiffExp, Collier, OpenLoops, ...

Amplitude → Master Integrals (MIs) → Evaluation of MIs



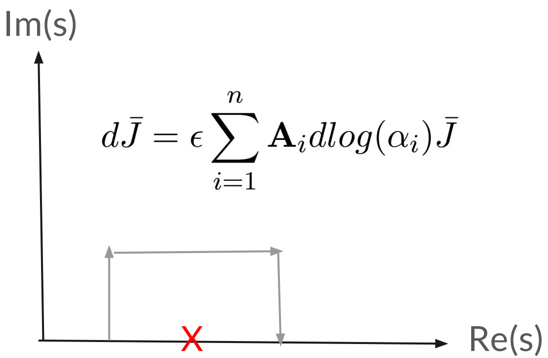


Motivation

Amplitude	Master Integrals (MIs)	Evaluation of MIs
Qgraf, Recola, FeynRules, FORM, FeynCalc, Tapir, …	Kira, Fire, Blade, LiteRed, NeatIBP, Reduze,...	AMFlow, pySecDec, DiffExp, Collier, OpenLoops, ...
		

What do we need?

- **Fast integrator** → No mathematica package
- Precise, but no need for 50 significant figures!
- Ideally, fast enough not to use a grid



$$d\bar{J} = \epsilon \sum_{i=1}^{n} \mathbf{A}_i dlog(\alpha_i) \bar{J}$$

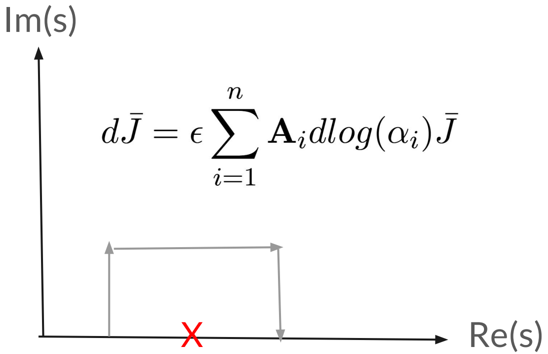
DiffExp [1] method: write Master Integrals in **differential form**, evolve it variable by variable from a boundary value to the desired final point with the **Frobenius method.** Avoid singularities with analytic continuation.

We could generate a grid of solutions with tools like DiffExp or SeaSyde [2], but dimensionality of the problem is large

[1] M. Hidding (2020)
[2] T. Armadillo et al (2022)

What do we need?

$$d\bar{J} = \epsilon \sum_{i=1}^{n} \mathbf{A}_i dlog(\alpha_i)\bar{J}$$



- **Fast integrator** → No mathematica package
- Precise, but no need for 50 significant figures!
- Ideally, fast enough not to use a grid

DiffExp [1] method: write Master Integrals in **differential form**, evolve it variable by variable from a boundary value to the desired final point with the **Frobenius method.** Avoid singularities with analytic continuation.

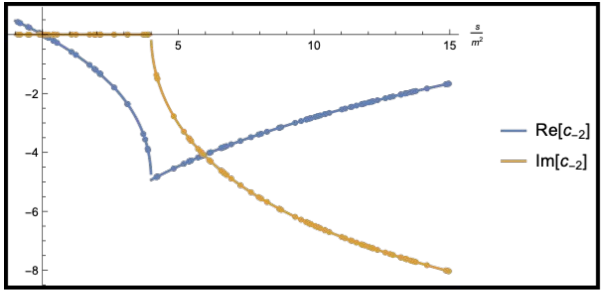
We could generate a grid of solutions with tools like DiffExp or SeaSyde [2], but dimensionality of the problem is large

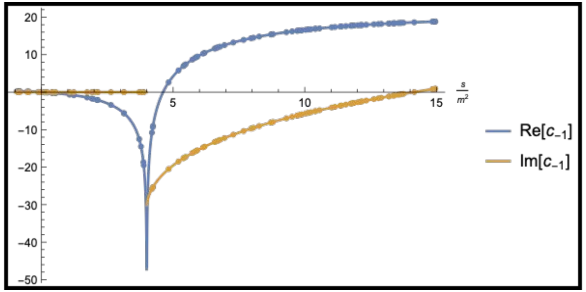
⟹ What if we evolve the differential equations **numerically**?

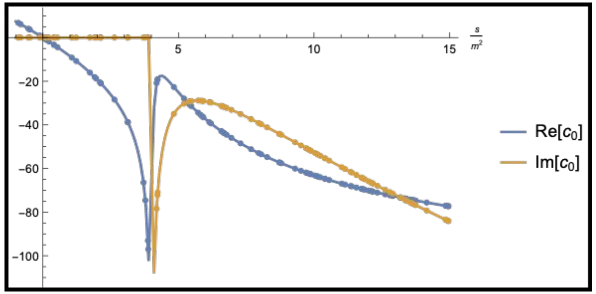
C++ integrator

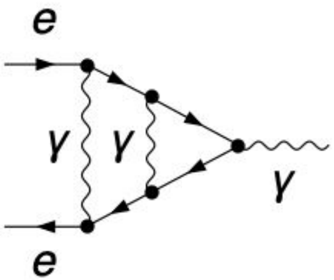
[1] M. Hidding (2020)
[2] T. Armadillo et al (2022)

Example



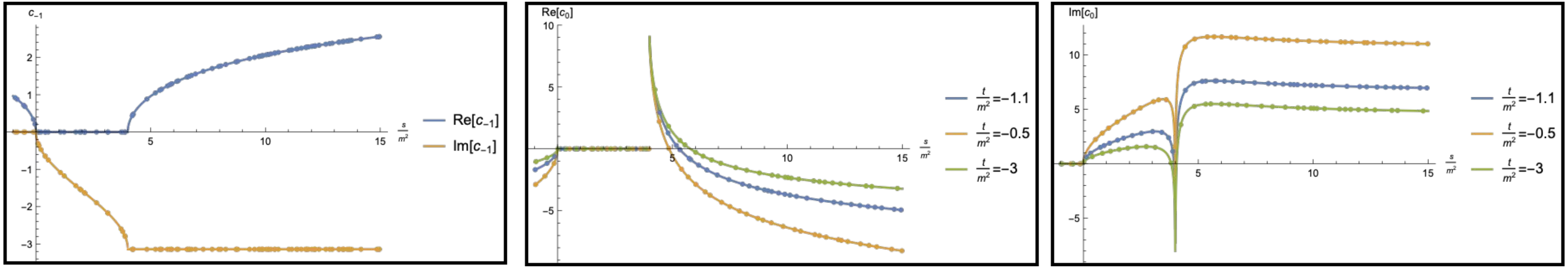


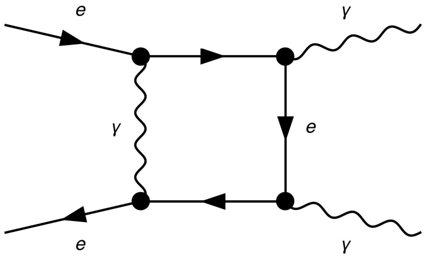




- 10 MIs, 5 orders of ε
- O(μs) per phase-space point
- 7+ significant figures of precision

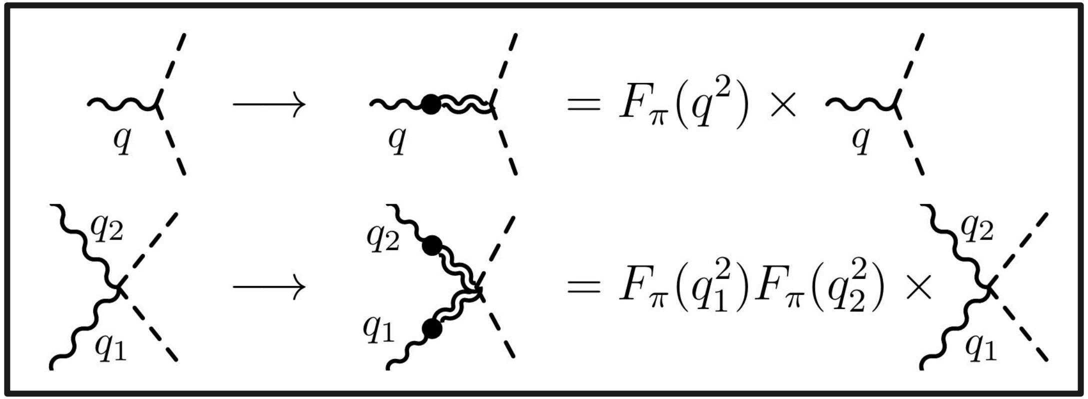
Example 2.0





- 8 MIs, 5 orders of ε
- O(*ms*) per phase-space point
- 7+ significant figures of precision

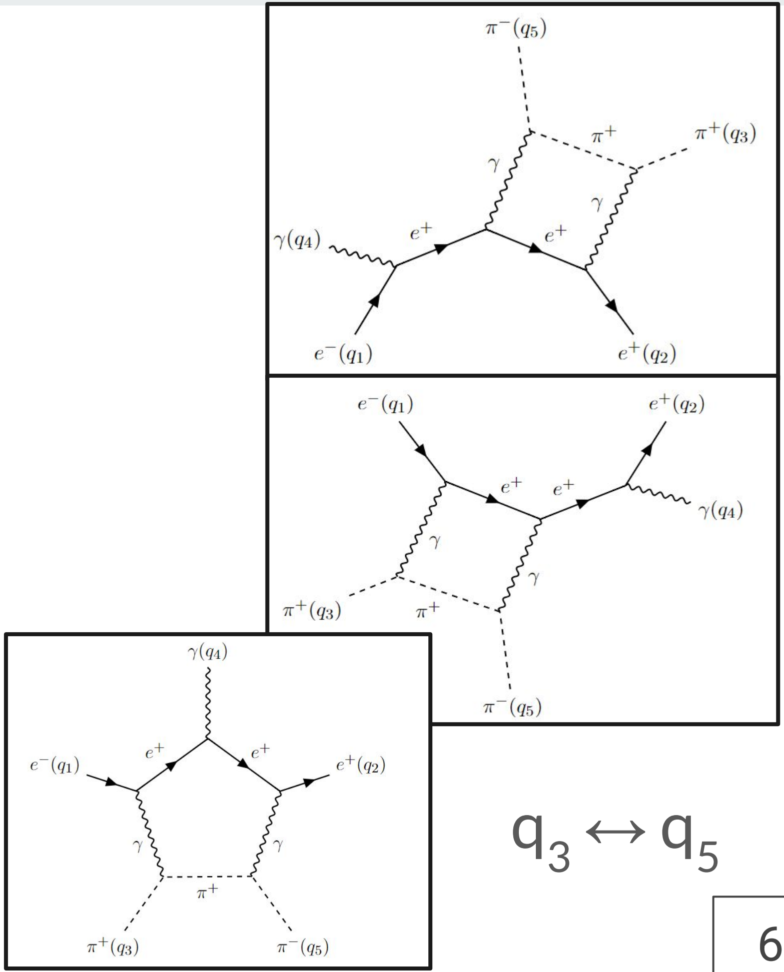
The GVMD model



$$F_\pi(q^2) = \sum_{v=1}^{3} a_v \frac{\Lambda_v}{\Lambda_v - q^2} = \sum_{v=1}^{3} a_v \left(1 + \frac{q^2}{\Lambda_v - q^2}\right)$$

Figure from F. Ignatov, R.N. Lee (2022)

ISR NLO 2γ* diagrams

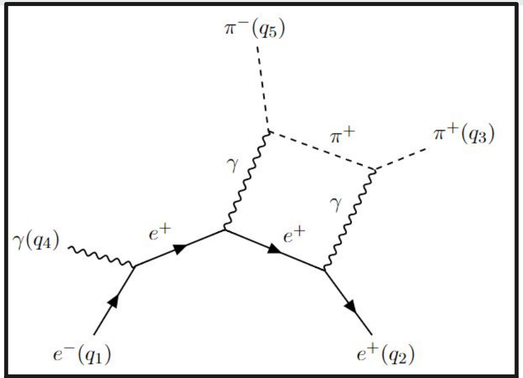


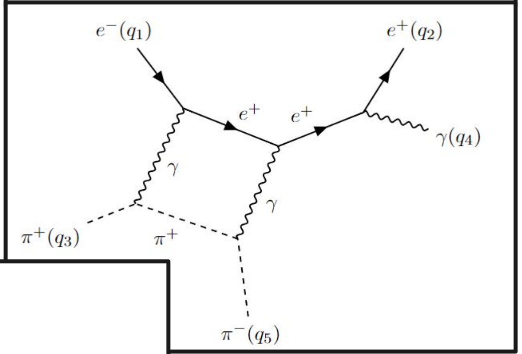
$q_3 \leftrightarrow q_5$

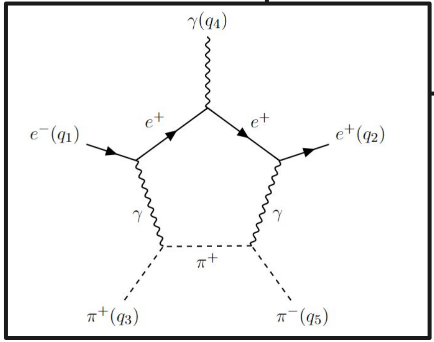
ISR NLO 2ɣ* diagrams

$$A_{TVP} \propto F_\pi(q_1^2) D_{\mu\nu}(q_1^2) F_\pi(q_2^2) D_{\mu\nu}(q_2^2)$$

$$A_{TVP} \propto \sum_{w=1}^{3} \sum_{v=1}^{3} a_v \left(D_{\mu\nu}(q_1^2) + \frac{i\eta_{\mu\nu}}{\Lambda_v - q_1^2} \right) a_w \left(D_{\mu\nu}(q_2^2) + \frac{i\eta_{\mu\nu}}{\Lambda_w - q_2^2} \right)$$







$q_3 \leftrightarrow q_5$

ISR NLO 2ɣ* diagrams

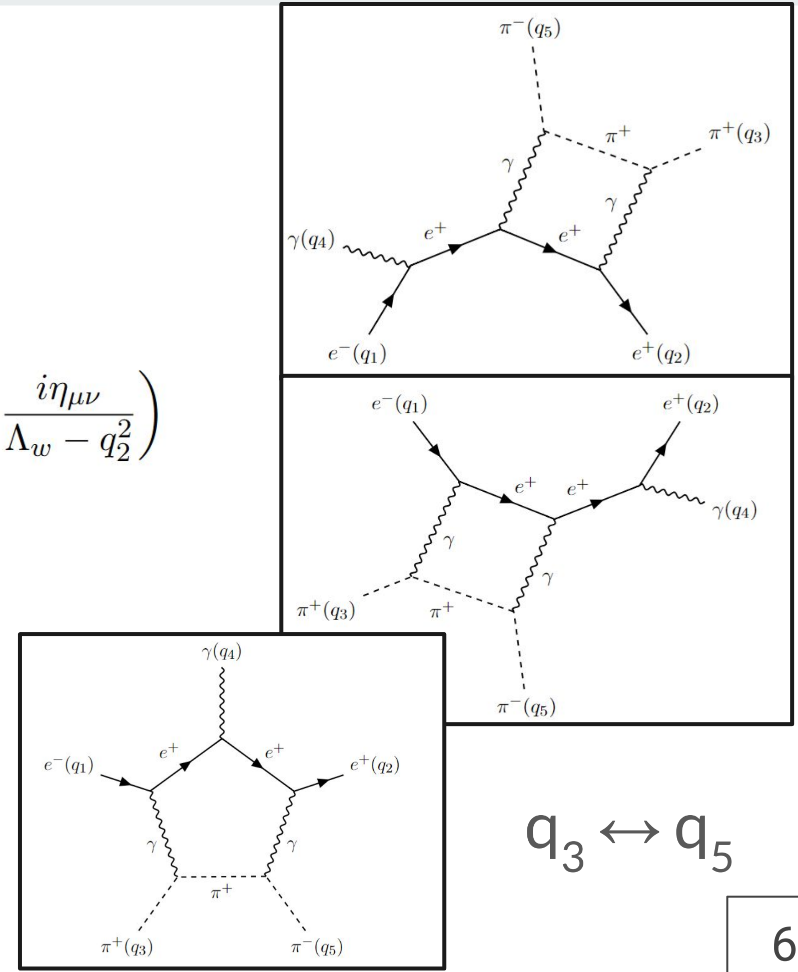
$$A_{TVP} \propto F_\pi(q_1^2) D_{\mu\nu}(q_1^2) F_\pi(q_2^2) D_{\mu\nu}(q_2^2)$$

$$A_{TVP} \propto \sum_{w=1}^{3} \sum_{v=1}^{3} a_v \left(D_{\mu\nu}(q_1^2) + \frac{i\eta_{\mu\nu}}{\Lambda_v - q_1^2} \right) a_w \left(D_{\mu\nu}(q_2^2) + \frac{i\eta_{\mu\nu}}{\Lambda_w - q_2^2} \right)$$

Still a 5 point integral at maximum, but we need to work with up to **9 kinematic variables**. We choose

$$\bar{x} = \{s_{14}, s_{15}, s_{23}, s_{24}, s_{35}, m_e^2, m_\pi^2, m_v^2, m_w^2\}$$

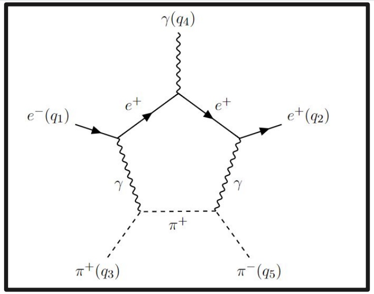
We have 16 possible combinations of m_v, m_w times 2 permutations of the external momenta



$q_3 \leftrightarrow q_5$

The topologies

$$I^X_{a,b,c,d,e} = \int \frac{d^Dk}{i\pi^{D/2}} \frac{1}{P_1^a P_2^b P_3^c P_4^d P_5^e}$$



	MM	MN	NN
P_1	$m_w^2 - p^2$	$-p^2$	$-p^2$
P_2	$m_\pi^2 - (p - q_3)^2$	$m_\pi^2 - (p - q_3)^2$	$m_\pi^2 - (p - q_3)^2$
P_3	$m_e^2 - (p + q_2)^2$	$m_e^2 - (p + q_2)^2$	$m_e^2 - (p + q_2)^2$
P_4	$m_v^2 - (p + q_{124})^2$	$m_v^2 - (p + q_{124})^2$	$-(p + q_{124})^2$
P_5	$m_e^2 - (p + q_{24})^2$	$m_e^2 - (p + q_{24})^2$	$m_e^2 - (p + q_{24})^2$

29 MIs 25 MIs 21 MIs

Obtaining the DE

- Canonical MIs obtained by studying Leading & Landau singularities in different dimensions
- Use of **FiniteFlow** [3] to reconstruct the DEs with an ansatz based on the **alphabet**

$$d\bar{J} = \epsilon \sum_{i=1}^{n} \mathbf{A}_i dlog(\alpha_i) \bar{J}$$

- Letters of the alphabet predicted by combining **BaikovLetters** [4] and **Effortless** [5]

$$\alpha_i = \frac{P(\bar{x}) + Q(\bar{x}) r_k}{P(\bar{x}) - Q(\bar{x}) r_k} \qquad \alpha_i = \frac{P(\bar{x}) + Q(\bar{x}) r_k r_j}{P(\bar{x}) - Q(\bar{x}) r_k r_j}$$

[3] T. Peraro (2019)
[4] X. Jiang (2024)
[5] A. Matijašić, J. Miczajka (xxxx)

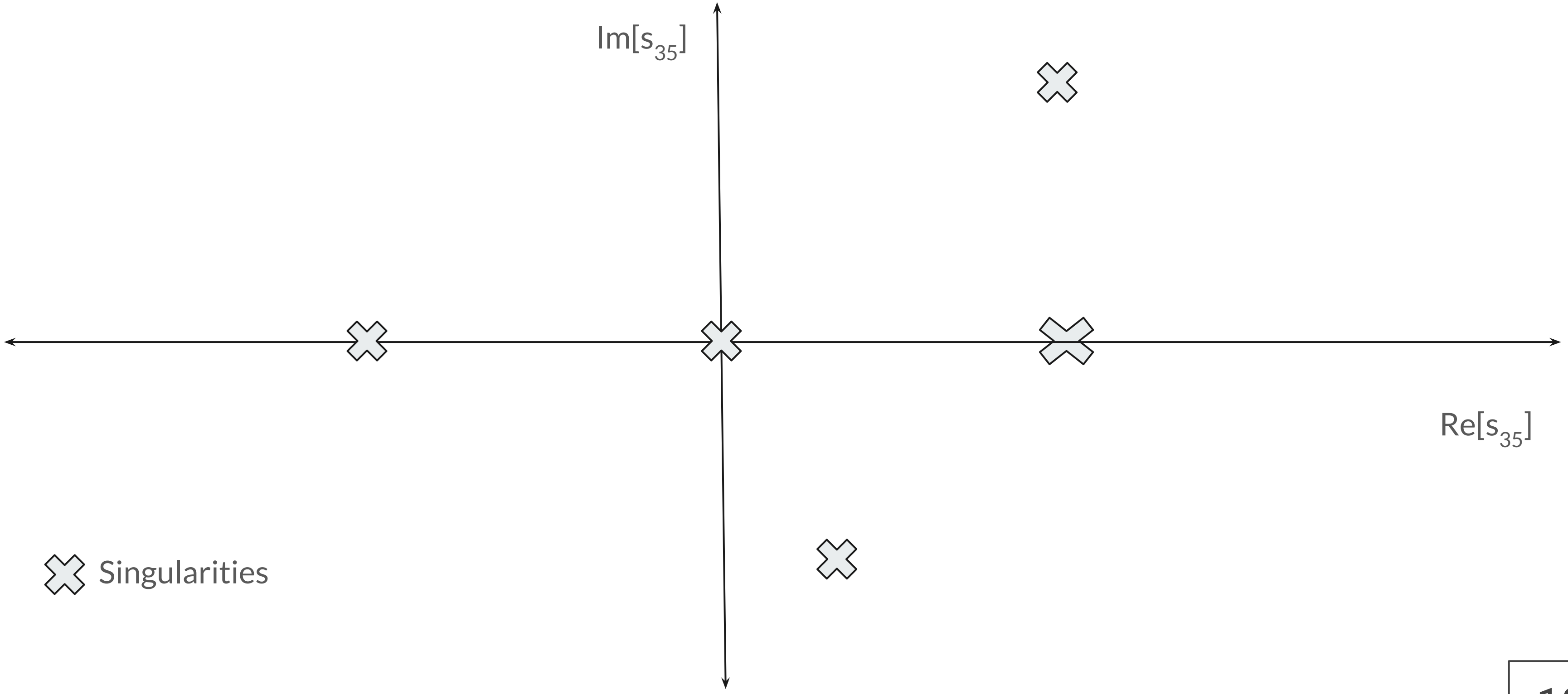
Returning to the integrator...

1. Get partial DE w.r.t. each kinematic variable
2. Input expression for each MI and order of epsilon in terms of letters and other MIs
3. Input values for pre-canonical MIs obtained with AMFlow at a non-singular arbitrary point
4. Input singularities and branch cuts
5. Input expressions for derivatives of letters and square roots
6. Find optimal path between origin and desired final point for each kin. var.
7. Evolve the DE variable by variable in that path:
    a. Multiply the AMFlow values by the canonical factors defined in terms of the current variable
    b. Solve the coupled partial DE with controlled stepper from **Boost Odeint** library
    c. Divide out the canonical factors from the solution
8. If desired, use the final result to go to a new final point

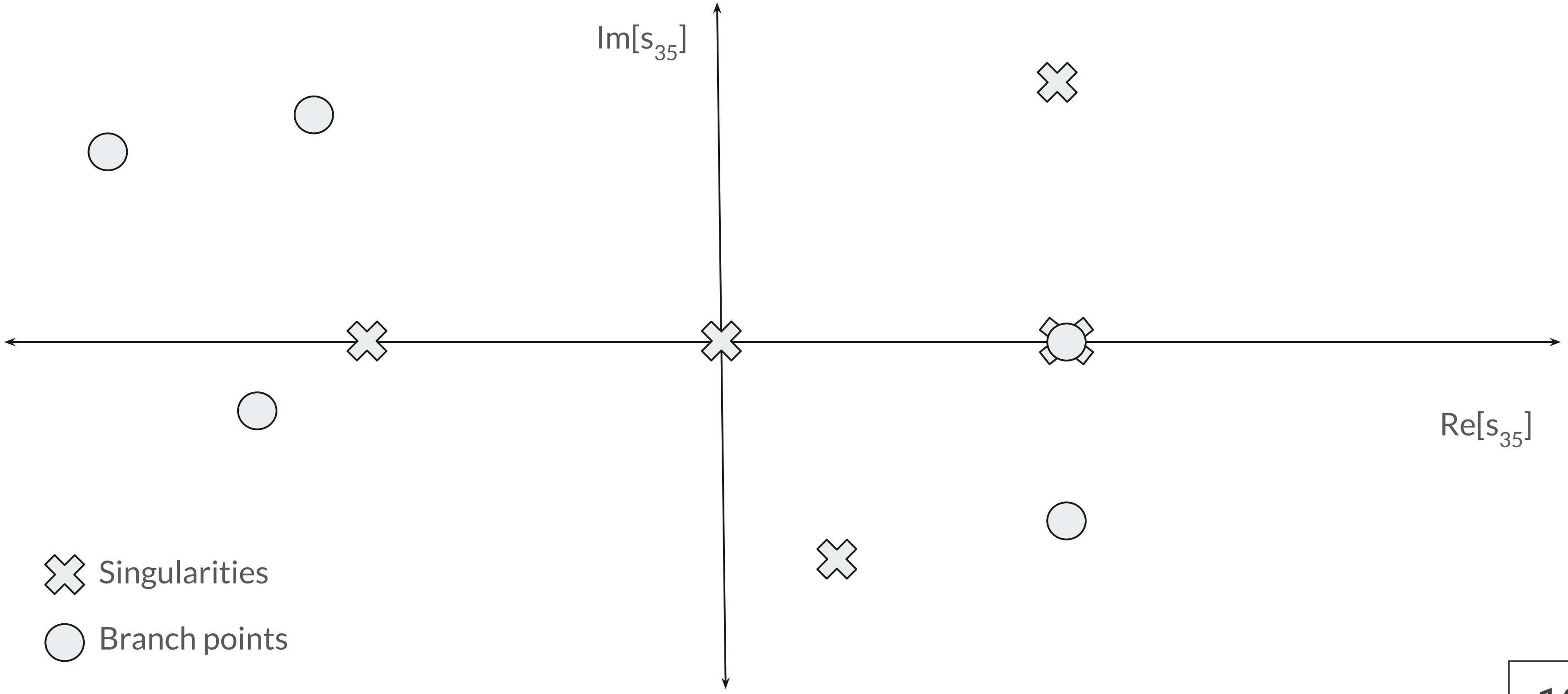
Returning to the integrator...

1. Get partial DE w.r.t. each kinematic variable
2. Input expression for each MI and order of epsilon in terms of letters and other MIs
3. Input values for pre-canonical MIs obtained with AMFlow at a non-singular arbitrary point
4. Input singularities and branch cuts
5. Input expressions for derivatives of letters and square roots
6. Find optimal path between origin and desired final point for each kin. var.
7. Evolve the DE variable by variable in that path:
    a. Multiply the AMFlow values by the canonical factors defined in terms of the current variable
    b. Solve the coupled partial DE with controlled stepper from **Boost Odeint** library
    c. Divide out the canonical factors from the solution
8. If desired, use the final result to go to a new final point

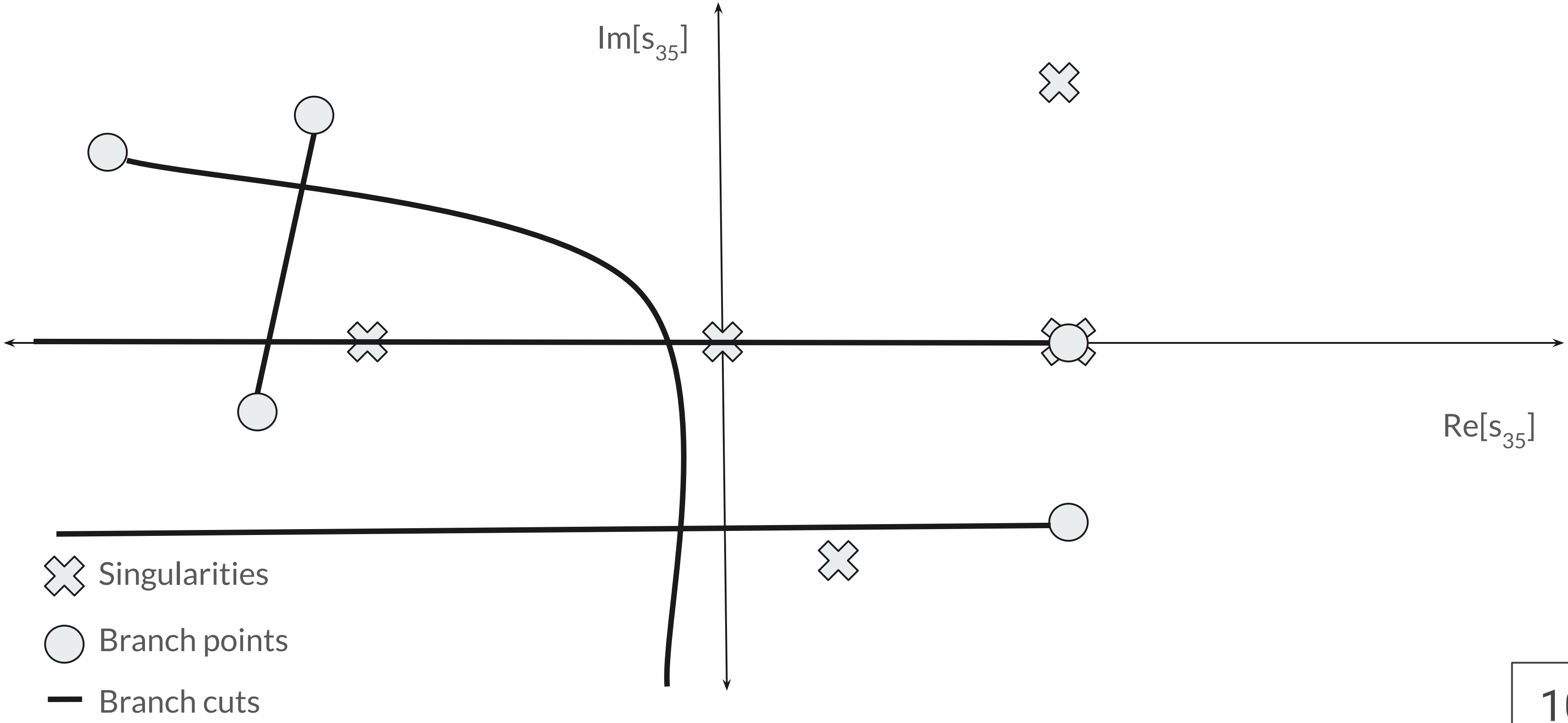
Branch cuts and paths



Branch cuts and paths



Branch cuts and paths



Branch cuts and paths

$r_i = \sqrt{K_i^n(\bar{x})} \Rightarrow K^n(\bar{x}) = \Pi_{j=1}^{n}(x - e_j) \Rightarrow \boxed{\Pi_{j=1}^{n}(x - e_j) - \lambda = 0}$

Branch cuts and paths

$$r_i = \sqrt{K_i^n(\bar{x})} \Rightarrow K^n(\bar{x}) = \Pi_{j=1}^n (x - e_j) \Rightarrow \boxed{\Pi_{j=1}^n (x - e_j) - \lambda = 0}$$

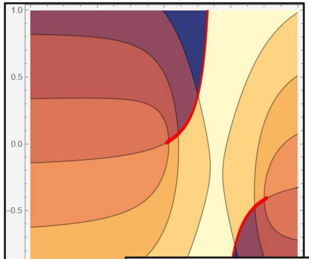
$$\Rightarrow r_i = \Pi_{j=1}^n \sqrt{x - e_j} \Rightarrow \boxed{e_j + \lambda_j = 0}$$

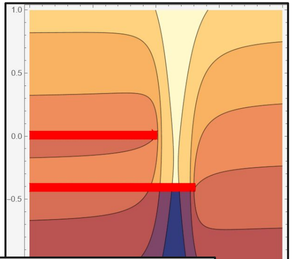
Branch cuts and paths

$$r_i = \sqrt{K_i^n(\bar{x})} \quad \Rightarrow \quad K^n(\bar{x}) = \Pi_{j=1}^n (x - e_j) \quad \Rightarrow \quad \boxed{\Pi_{j=1}^n (x - e_j) - \lambda = 0}$$

$$r_i = \Pi_{j=1}^n \sqrt{x - e_j} \quad \Rightarrow \quad \boxed{e_j + \lambda_j = 0}$$

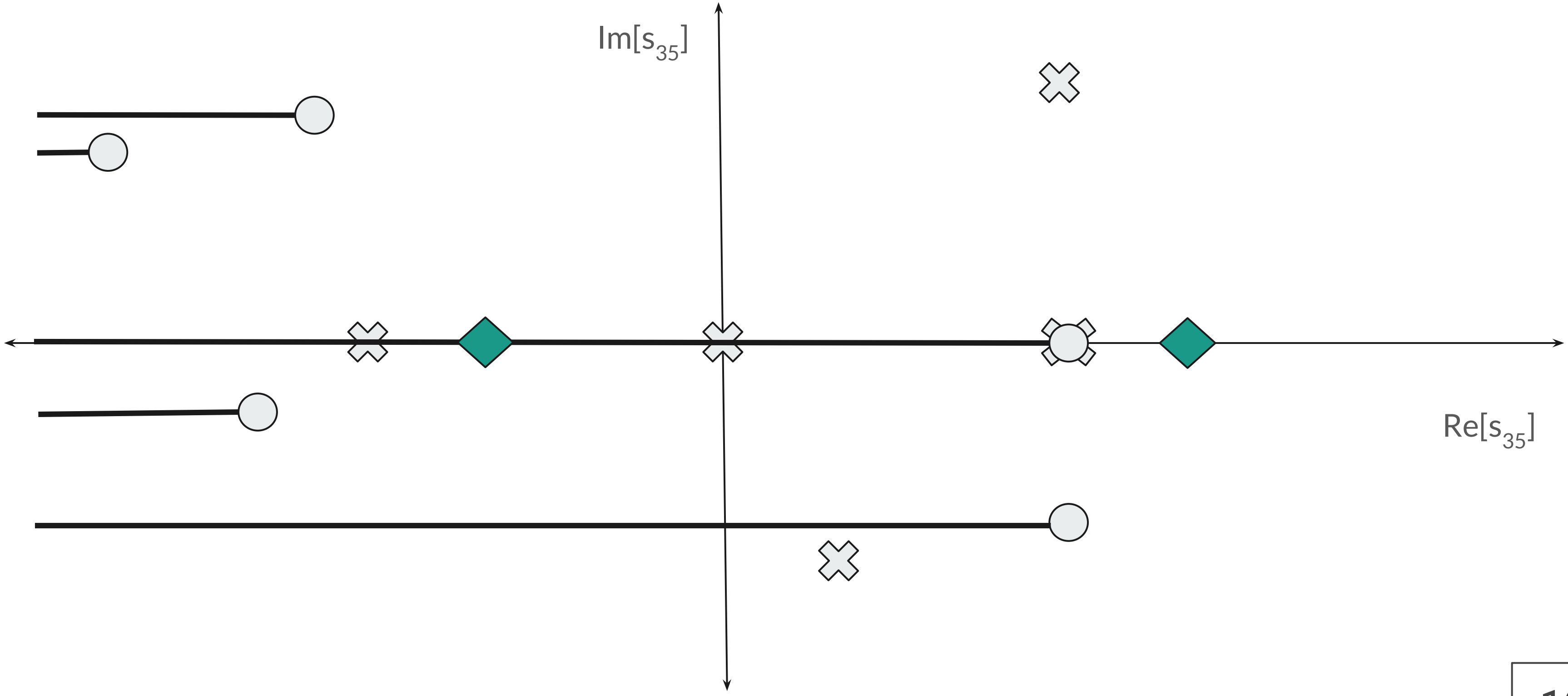
We use the standard convention from mathematical software: branch cuts parallel to the negative real axis.



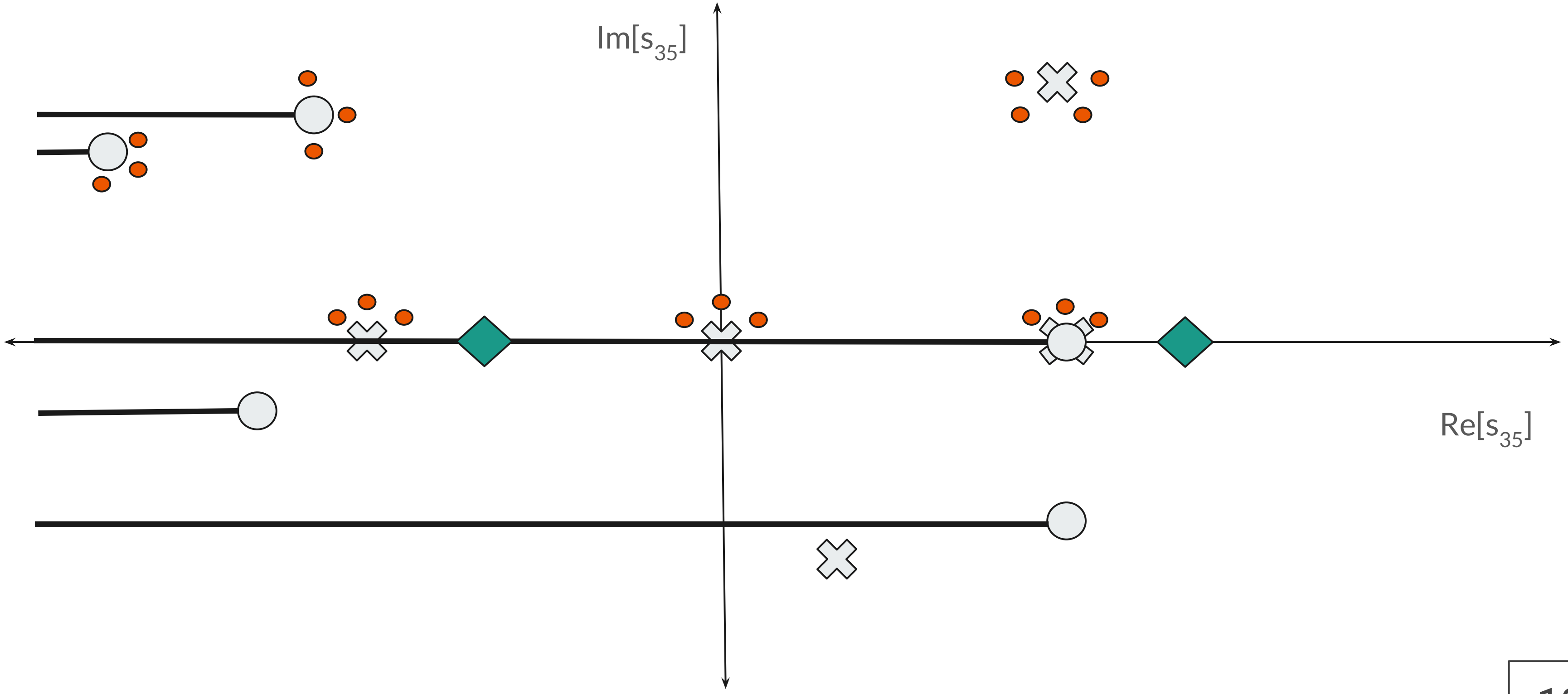


$$r_{14} = \sqrt{m_v^4 + (m_w^2 - s_{35})^2 - 2m_v^2(m_w^2 + s_{35})}$$

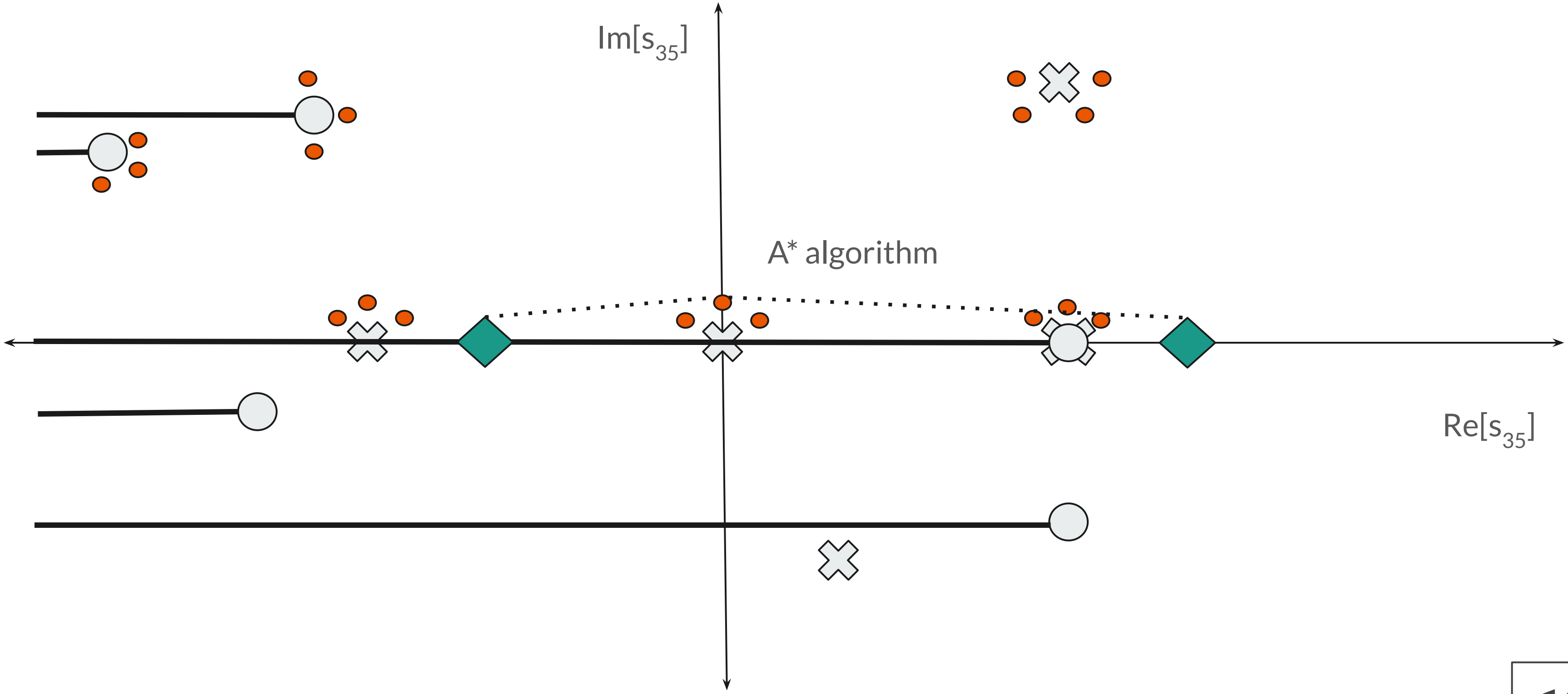
Branch cuts and paths



Branch cuts and paths



Branch cuts and paths



More analytic manipulation....

For the NN topology, more simplifications:

- No need for **21** MIs **x 3** O(ε) **x 2** q(5) ↔ q(3)
- Use a rotation matrix R to evaluate only the relevant functions

$$\vec{W}(\vec{x};\epsilon) = R\,\vec{J}(\vec{x};\epsilon)$$

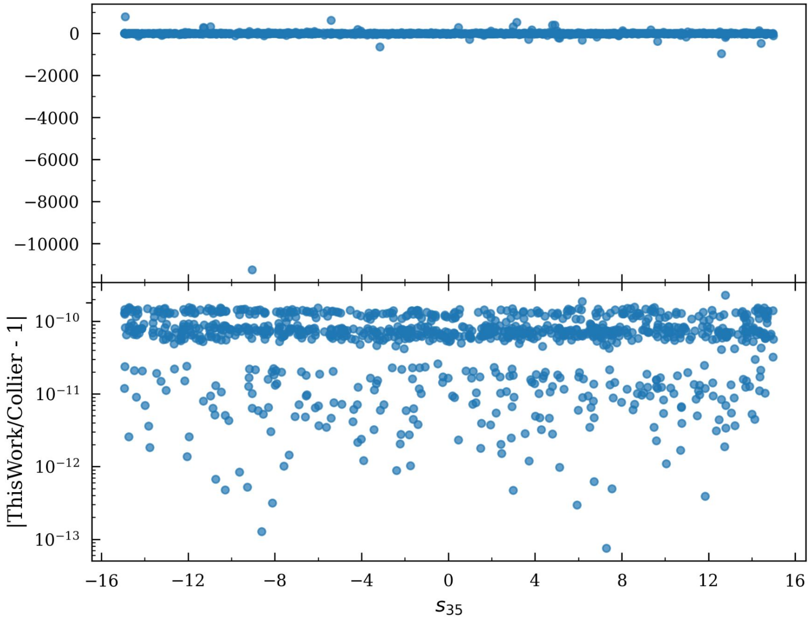
- By decomposing in terms of independent functions,

118 MIs → 65 Ws

This can be applied to MN and MM, but less simplification is expected

Check for the Ws

Comparison with Collier. Results of $I_{1,0,1,1,1}$ at finite order

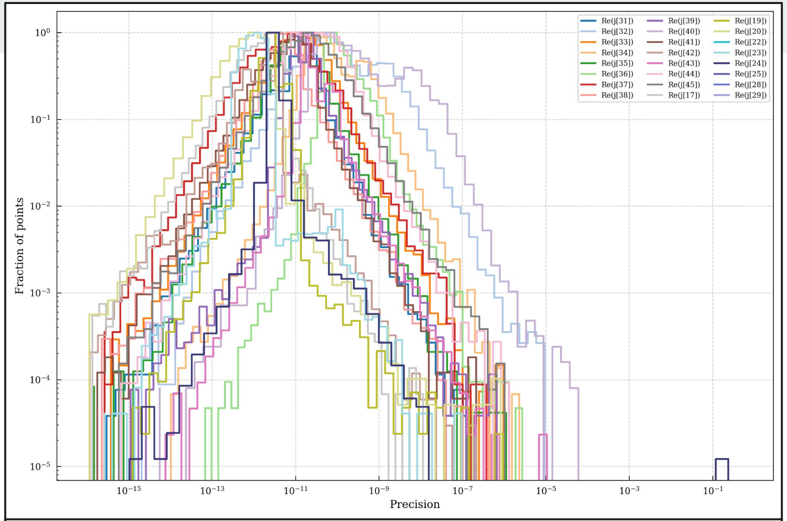


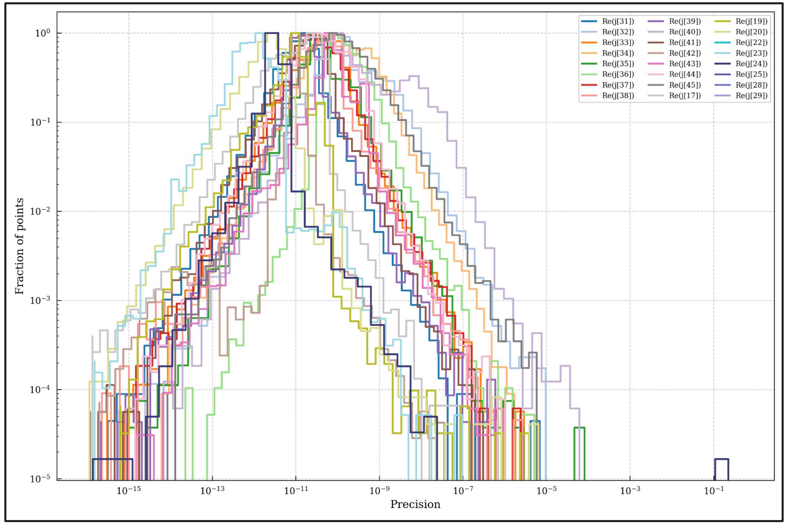
*Thanks to Daniel Gerardo Melo Porras

Precision for MM

Complex mass = x

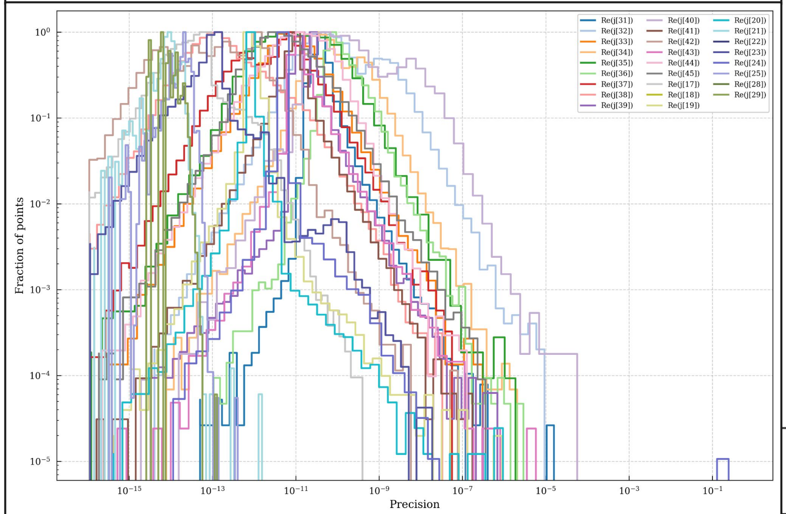
Runge-Kutta
Dopri





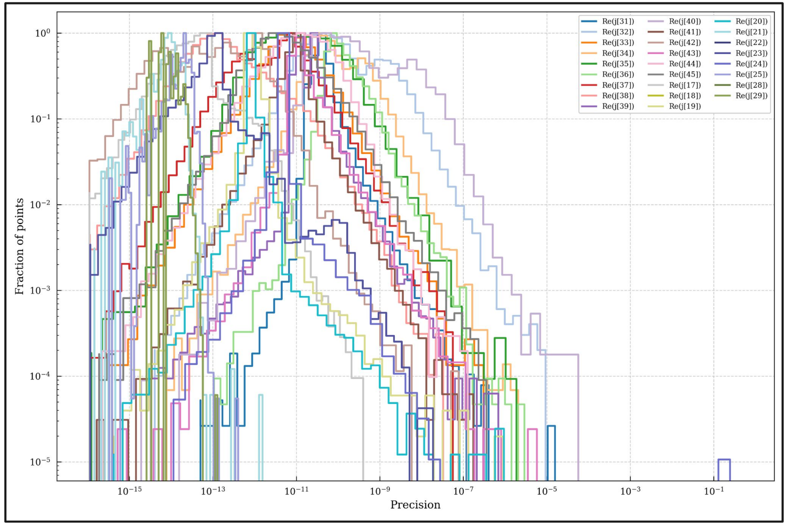
Runge-Kutta Cash Karp 45

Bulirsch-Stoer



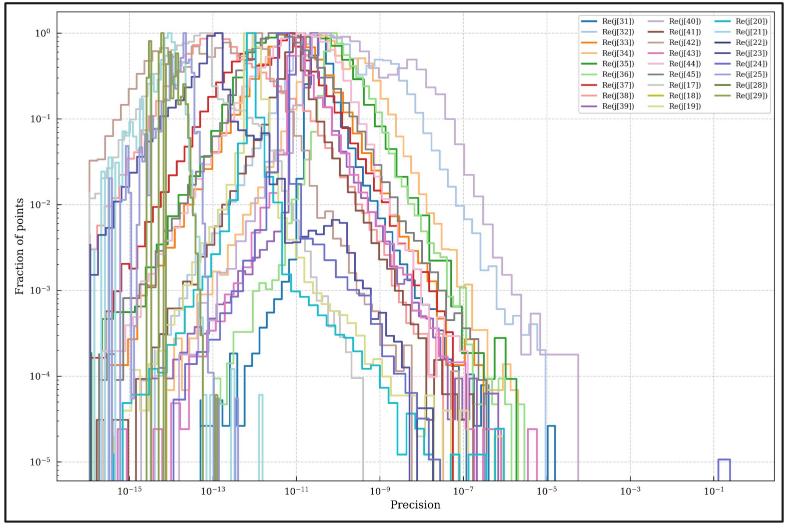
Precision for MM

Complex mass = x

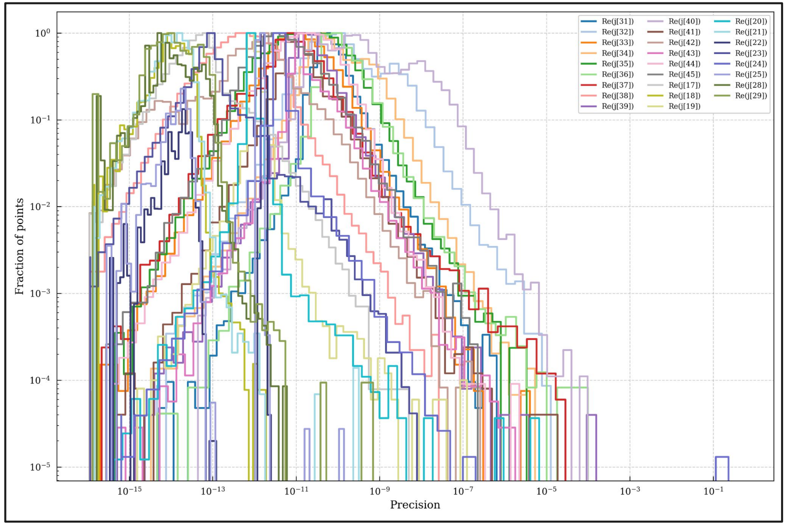


Precision for MM

Complex mass = x

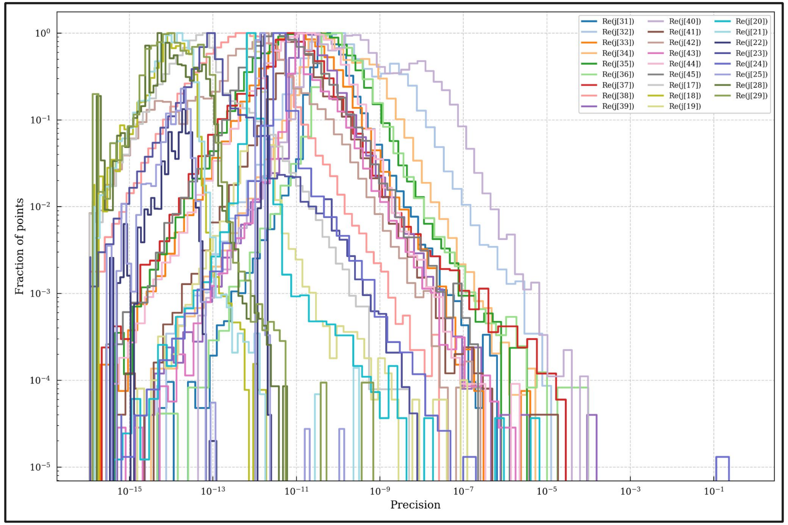


Complex mass = x, y, z

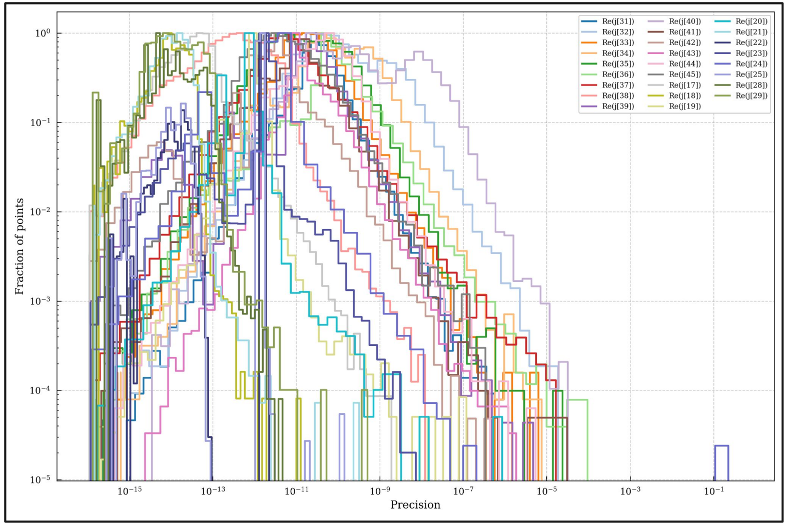


Precision for MM

Complex mass = x, y, z

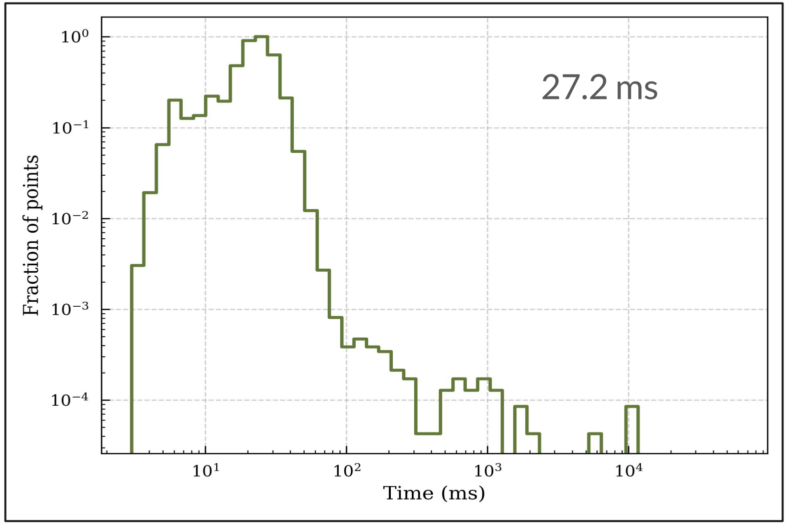


Complex mass = x, y, z & 6 b.c



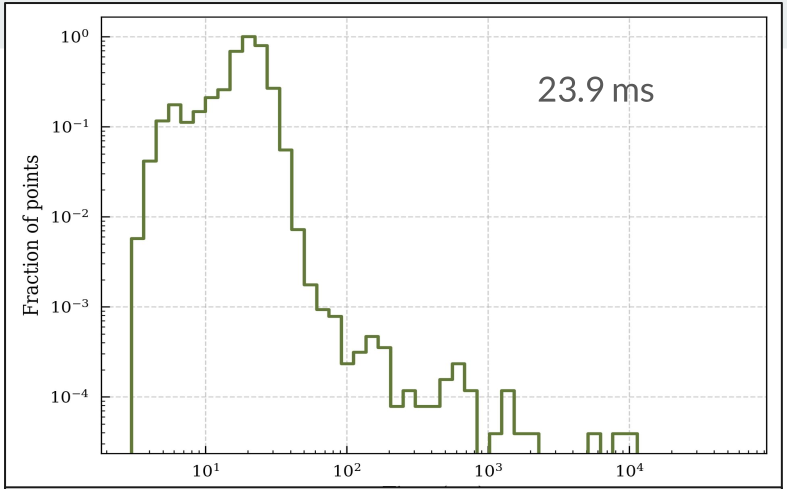
Runtime for MM

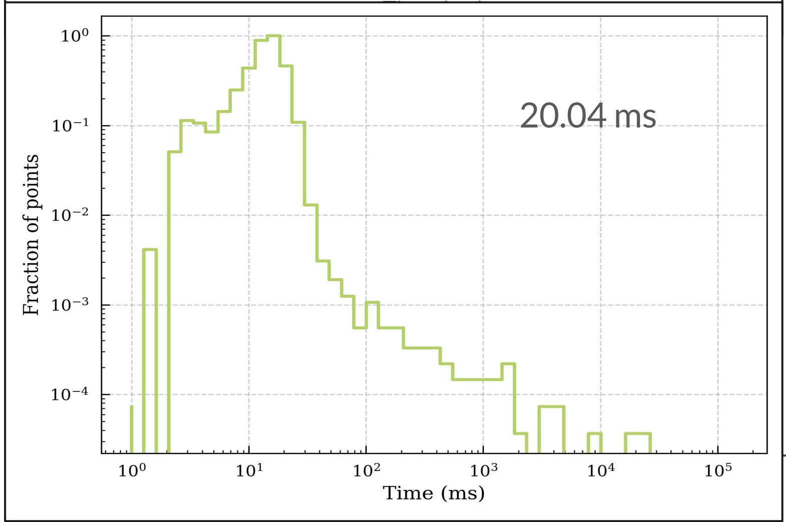
Complex mass = x



Runge-Kutta Cash Karp 45

Runge-Kutta Dopri

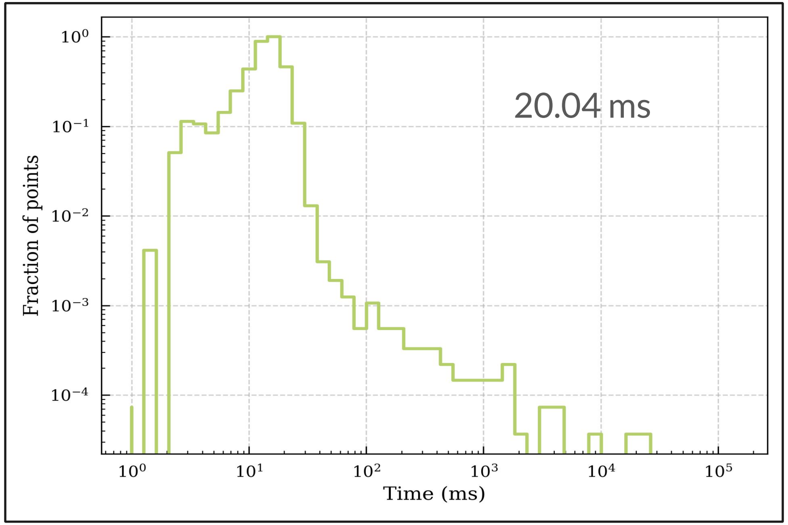




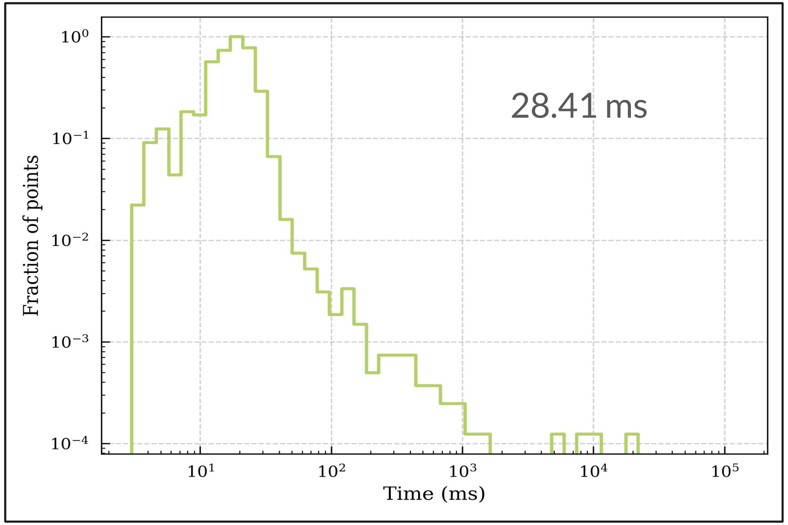
Bulirsch-Stoer

Runtime for MM

Complex mass = x

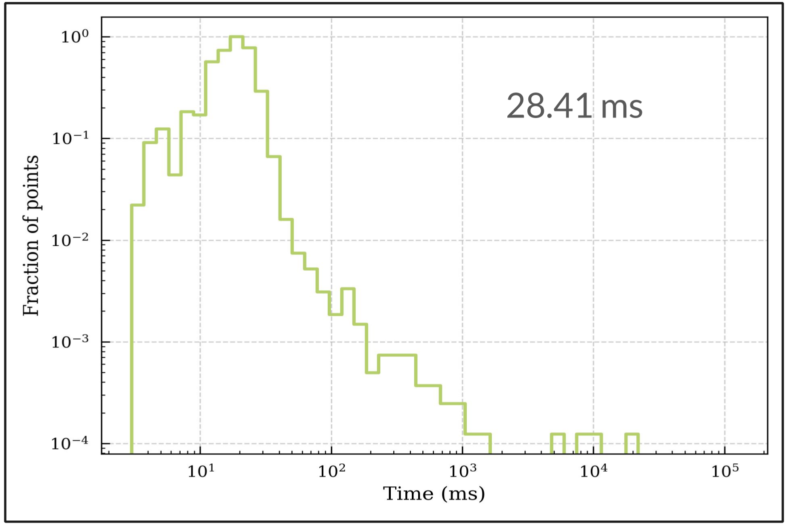


Complex mass = x, y, z

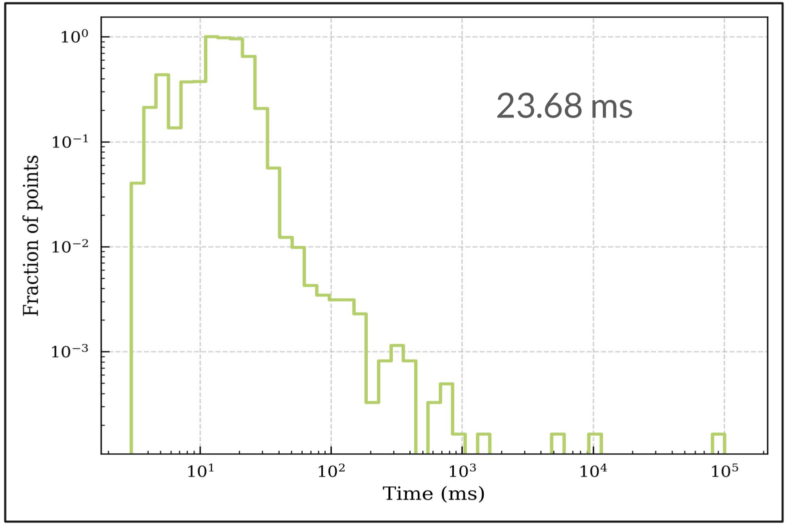


Runtime for MM

Complex mass = x, y, z



Complex mass = x, y, z & 6 b.c.



Next steps

1. Decompose amplitude in Form Factors
2. Implement in Phokhara, also GVMD amplitude (compare time with Collier)
3. Need for quad. prec. version to check error in higher orders of ε
4. Generate small ~100 boundary values grid
5. Explore implicit methods for solving DEs, maybe exponential integrator?
6. Solving DEs numerically is a huge field, learn from them
7. NNLO $\rightarrow$ See W. T. Bobadilla's talk

Next steps

1. Decompose amplitude in Form Factors
2. Implement in Phokhara, also GVMD amplitude (compare time with Collier)
3. Need for quad. prec. version to check error in higher orders of ε
4. Generate small ~100 boundary values grid
5. Explore implicit methods for solving DEs, maybe exponential integrator?
6. Solving DEs numerically is a huge field, learn from them
7. NNLO $\rightarrow$ See W. T. Bobadilla's talk

Thanks!