Jonathan Ronca

LoopIn: Loop Integrals

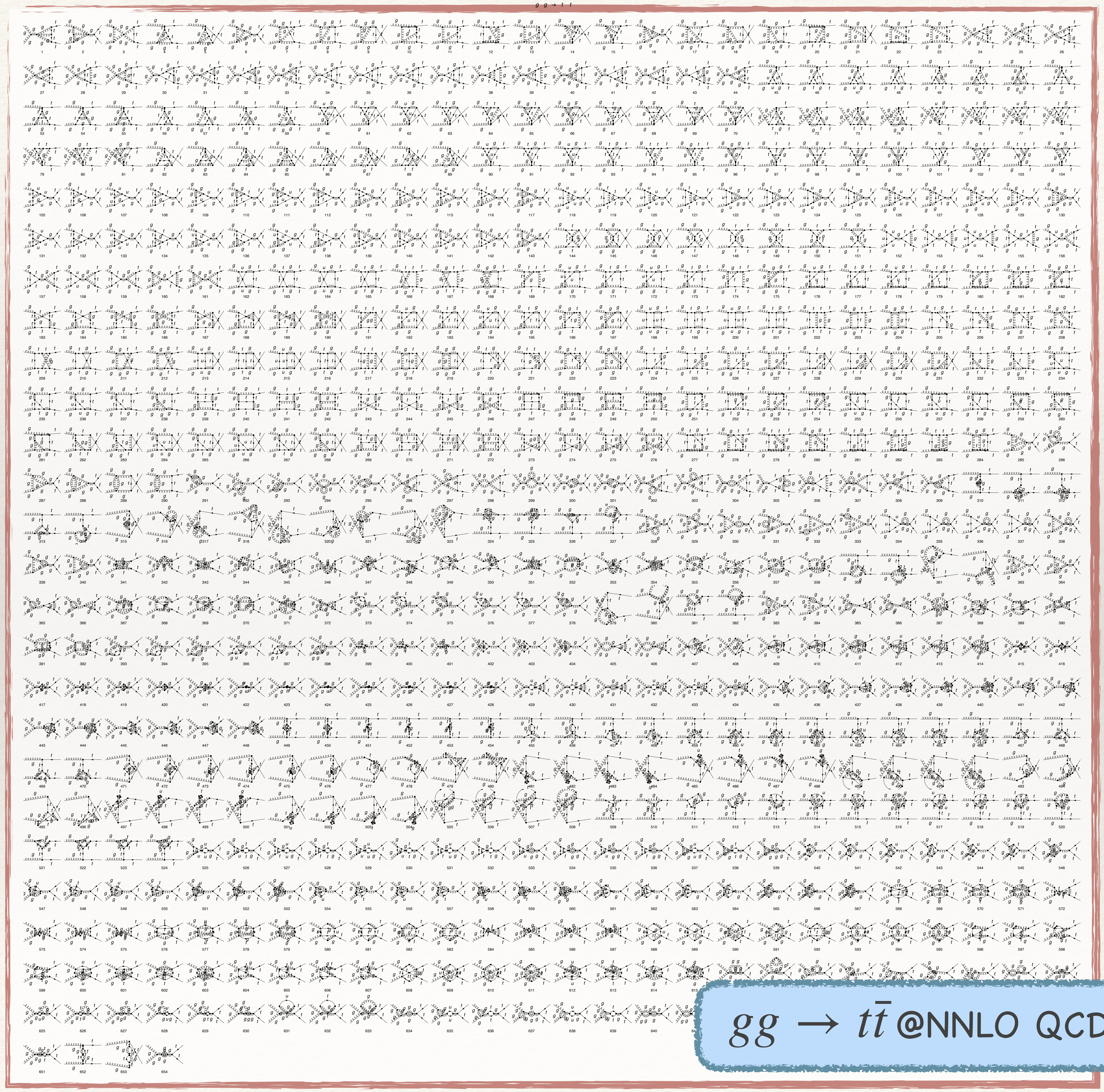
In collaboration with: ***M. Bigazzi, M. K. Mandal, P. Mastrolia, W. J. Torres Bobadilla***

Dipartimento di Fisica e Astronomia Galileo Galilei

INFN PADOVA
Istituto Nazionale di Fisica Nucleare
Sezione di Padova

11 Dec 2024

Motivation



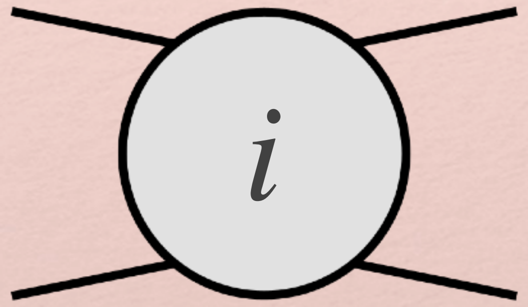
$gg \to t\bar{t}$ @NNLO QCD

pQFT = Scattering amplitudes are expanded w.r.t the coupling

$$\mathscr{A} = \sum \left(\frac{\alpha}{\pi}\right)^n \mathscr{A}^{(n)}$$

n-th term = N^nLO

$$\mathscr{A}^{(n)}|_{\text{virt}} = \sum_{i=1}^{\#\text{diags}}$$



n = number of loop

For typical NNLO QCD amplitudes #diags = O(10^2 - 10^3)

For each diagram O(10^2) integrals

How can we compute amplitudes systematically?

Motivation

Flagship Use Cases [UC2.1.3] : Advanced Calculus for Precision Physics

LOOPIN
Loop Integrals

A completely automated code for numerical evaluation of Scattering Amplitude

What we are aiming for LoopIn to be?

- Mathematica front-end (user-friendly)
- Minimal number of inputs
- From the generation of the amplitude to its numerical evaluation
- Modular: LoopIn has to be able to be interfaced with any code
- Flexible: User can manipulate the IO of LoopIn (with care)
- Every module of LoopIn will produce its own output
- Parallelizable
- Designed for any number of loop

LoopIn: where are we now?

KPI ID	Description	Acceptance threshold	To be obtained by
KPI 2.1.3.1	Identification of open-source codes and libraries	Name of the open-source codes	MS6 ✓
KPI 2.1.3.2	Deployment of working code for scattering amplitude evaluation	First version of code and libraries, proof of concept, running on single core	MS9 ✓
KPI 2.1.3.3	Code and libraries optimization	Improved CPU performances to be verified against known result for particle scattering within SM, for 2-to-2 scattering amplitudes	MS9
KPI 2.1.3.4	Delivery of the final (MPI/openMP) code and libraries for scattering amplitude evaluations	Final version of the code and libraries running on at least 48 cores; simulations and benchmarks on the Leonardo HPC system, for processes 2-to-3 scattering	MS12

Inside LoopIn

LoopIn: setting up the calculation

How do we setup the calculation of each contribution to the cross section?

Process

$\gamma^* \to \bar{l} l \gamma$

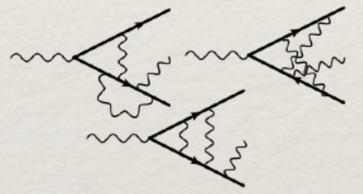
$\mathrm{N^2LO\ QED}$

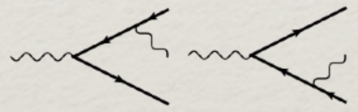
$(\mathrm{tree}, 2L)$ – interference

$\mathscr{A}_p^n =$ p-point n-loop Feynman amplitude

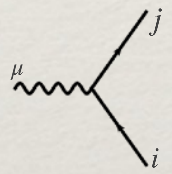
$(\mathscr{M}_p^n)_{ij} =$ Interference between i-th $\mathscr{A}_p^{(0)}$ term And j-th $\mathscr{A}_p^{(n)}$ term

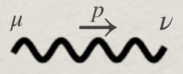
Drawing Feynman Diagrams

$\mathscr{A}_4^2 =$  $\cdots$

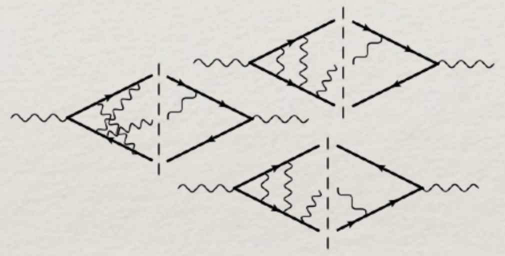
$\mathscr{A}_4^0 =$ 

Feynman Rules

 $\to ie\gamma^\mu \delta_{ij}$

 $\to \dfrac{-ig^{\mu\nu}}{p^2 - i\epsilon}$

Building Interference terms

$\mathscr{M}_4^2|_{\mathrm{virt}} =$ 

Summing over external states

$$\sum_h \epsilon_i^{(h)*} \epsilon_j^{(h)} \to g^{\mu\nu}$$

$$\sum_s \bar{\psi}^{(s_j)}(p_j)\cdots\psi^{(s_i)}(p_i)\bar{\psi}^{(s_i)}(p_i)\cdots\psi^{(s_j)}(p_j) \to \mathrm{Tr}(\not{p}_j\cdots\not{p}_i\cdots)$$

Numerator Algebra

$$p_i^\mu p_j^\nu g_{\mu\nu} = p_i \cdot p_j$$

$$\mathrm{Tr}(\cdots) = \sum_{ij} c_{ij}(p_i \cdot p_j)$$

Interference = combination of Integrals

$$(\mathscr{M}_p^n)_{ij} = \sum_{\bar{\alpha}_n \bar{\beta}_r} c_{\bar{\alpha}_n \bar{\beta}_r} I_{\bar{\alpha}_n}^{\bar{\beta}_r}\left(\boldsymbol{s}, \boldsymbol{m}^2; d\right)$$

Exporting all interference terms

Feynman Integrals

$$I_{\bar{\alpha}_n}^{\bar{\beta}_r}\left(\boldsymbol{s}, \boldsymbol{m}^2; d\right) = \int \prod_{l=1}^{L} d^d k_l \frac{D_{n+1}^{\beta_1} \cdots D_{n+r}^{\beta_r}}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

$D_i = q_i^2 - m_i^2 + i\epsilon$

q_i is a combination of loop and external momenta

- Each interference term can be expressed as a linear combination of Feynman Integrals
- Feynman integrals depend will depend on invariants, masses and space-time dimension*

*dimensional regularization is implicitly assumed during the whole discussion

Integration-by-parts identities

Is there a way to reduce the number of integrals we need to evaluate?

Typical 2-loop processes get contribution from $\sim 10^4 - 10^6$ integrals

Feynman Integrals

$$I^{\bar{\beta}_r}_{\bar{\alpha}_n}\left(s, m^2; d\right) = \int \prod_{l=1}^{L} d^d k_l \frac{D_{n+1}^{\beta_1} \cdots D_{n+r}^{\beta_r}}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

Fact: Feynman integrals are invariant over loop momenta shifts

Integration-by-part Identities (IBPs)

$$k_l \to A_{li} k_i + B_{lj} p_j \implies \int \prod_{l=1}^{L} d^d k_l \frac{d}{dk_j^{\mu}} \left(v_i^{\mu} \frac{D_{n+1}^{\beta_1} \cdots D_{n+r}^{\beta_r}}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}} \right) = 0$$

[Chetyrkin,Tkachov:1981]
[Laporta:hep-ph/0102033]

Using $\bar{\alpha}_r, \bar{\beta}_s$ as seeds: GIGANTIC system of equation

$$\sum_{\bar{\alpha}_s, \bar{\beta}_r} b_{\bar{\alpha}_s, \bar{\beta}_r} \mathrm{Int}(T, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_r) = 0$$

```
-Int[T, 1, 0, 0, 0, 0, -1, 0, 0, 0] α0 +
  Int[T, 1, 0, 0, 0, 0, 0, 0, -1, 0] α0 + Int[T, -1, 1, 0, 0, 0, 0, 0, 0, 0] α1 +
  Int[T, 0, 1, 0, 0, -1, 0, 0, 0, 0] α1 - Int[T, 0, 1, 0, 0, 0, -1, 0, 0, 0] α1 +
  Int[T, -1, 0, 1, 0, 0, 0, 0, 0, 0] α2 + Int[T, 0, 0, 1, -1, 0, 0, 0, 0, 0] α2 -
  Int[T, 0, 0, 1, 0, 0, -1, 0, 0, 0] α2 + Int[T, 0, 0, 0, 0, 0, 0, 0, 0, 0] (α0 - α5) +
  Int[T, -1, 0, 0, 0, 0, 1, 0, 0, 0] α5 - Int[T, 0, 0, 0, 0, 0, 1, 0, -1, 0] α5 +
  Int[T, -1, 0, 0, 0, 0, 0, 0, 0, 1] β8 - Int[T, 0, 0, 0, -1, 0, 0, 0, 0, 1] β8 -
  Int[T, 0, 0, 0, 0, 0, -1, 0, 0, 1] β8 + Int[T, 0, 0, 0, 0, 0, 0, -1, 0, 1] β8 +
  Int[T, 0, 0, 0, 0, 0, 0, 0, -1, 1] β8 - t Int[T, 0, 0, 0, 0, 0, 0, 0, 0, 1] β8 == 0
```

Example of an IBP operator

Integration-by-parts identities

Interference

$$(\mathscr{M}_p^{(L)})_{ij} = \sum_{\bar{\alpha}_n \bar{\beta}_r} c_{\bar{\alpha}_n \bar{\beta}_r} \, I_{\bar{\alpha}_n}^{\bar{\beta}_r}(\boldsymbol{s}, \boldsymbol{m}^2; d)$$

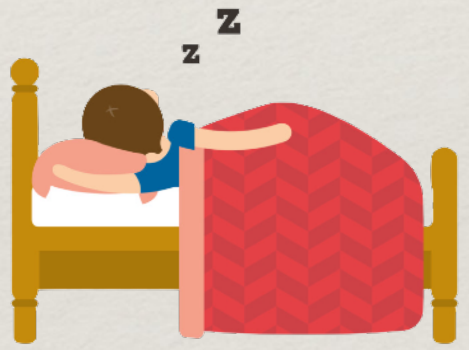
Reading Integrals From Interferences

```
Int[T, -1, 1, 2, 1, 1, 0, 0, 0, 0]
Int[T, -1, 2, 0, 1, 0, 1, 0, 0, 0]
Int[T, -1, 2, 1, 1, 1, 0, 0, 0, 0]
Int[T, 0, 1, 0, 1, 0, 1, 0, 0, 0]
Int[T, 0, 1, 0, 2, 0, 1, 0, -1, 0]
Int[T, 0, 1, 1, 0, 2, 0, 0, 0, 0]
Int[T, 0, 1, 1, 1, 1, 0, 0, 0, 0]
Int[T, 0, 1, 1, 1, 2, 0, -1, 0, 0]
Int[T, 0, 1, 1, 1, 2, 0, 0, -1, 0]
Int[T, 0, 1, 1, 1, 2, 0, 0, 0, 0]
Int[T, 0, 1, 1, 2, 1, 0, 0, -1, 0]
Int[T, 0, 1, 2, 0, 0, 1, 1, 0, 0]
Int[T, 0, 1, 2, 1, 1, 1, 1, 0, 0]
Int[T, 0, 1, 2, 1, 1, 1, 2, 0, 0]
Int[T, 0, 2, -1, 1, 0, 1, 0, 0, 0]
Int[T, 0, 2, 0, 1, 0, 1, 0, 0, -1]
```

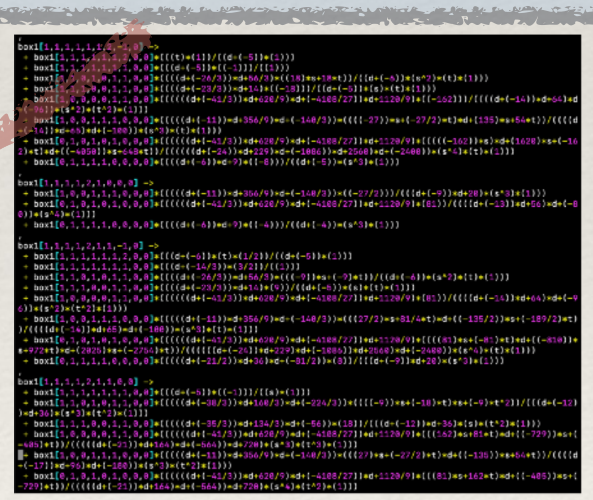
Creating Inputs for the IBP reducer

```
integralfamilies:
  - name: "T"
    loop_momenta: [k1,k2]
    top_level_sectors: [127]
    propagators:
      - [ k1, m2 ]
      - [ k1-p1, m2 ]
      - [ k1+p2, 0 ]
      - [ k2-p2, m2 ]
      - [ k2+p1, 0 ]
      - [ k1+k2, m2 ]
      - [ k2-p2-p3, m2 ]
      - [ k2, 0 ]
      - [ k1 + p3, 0 ]
```

Running IBPs (Possibly in parallel)



Reducer Output

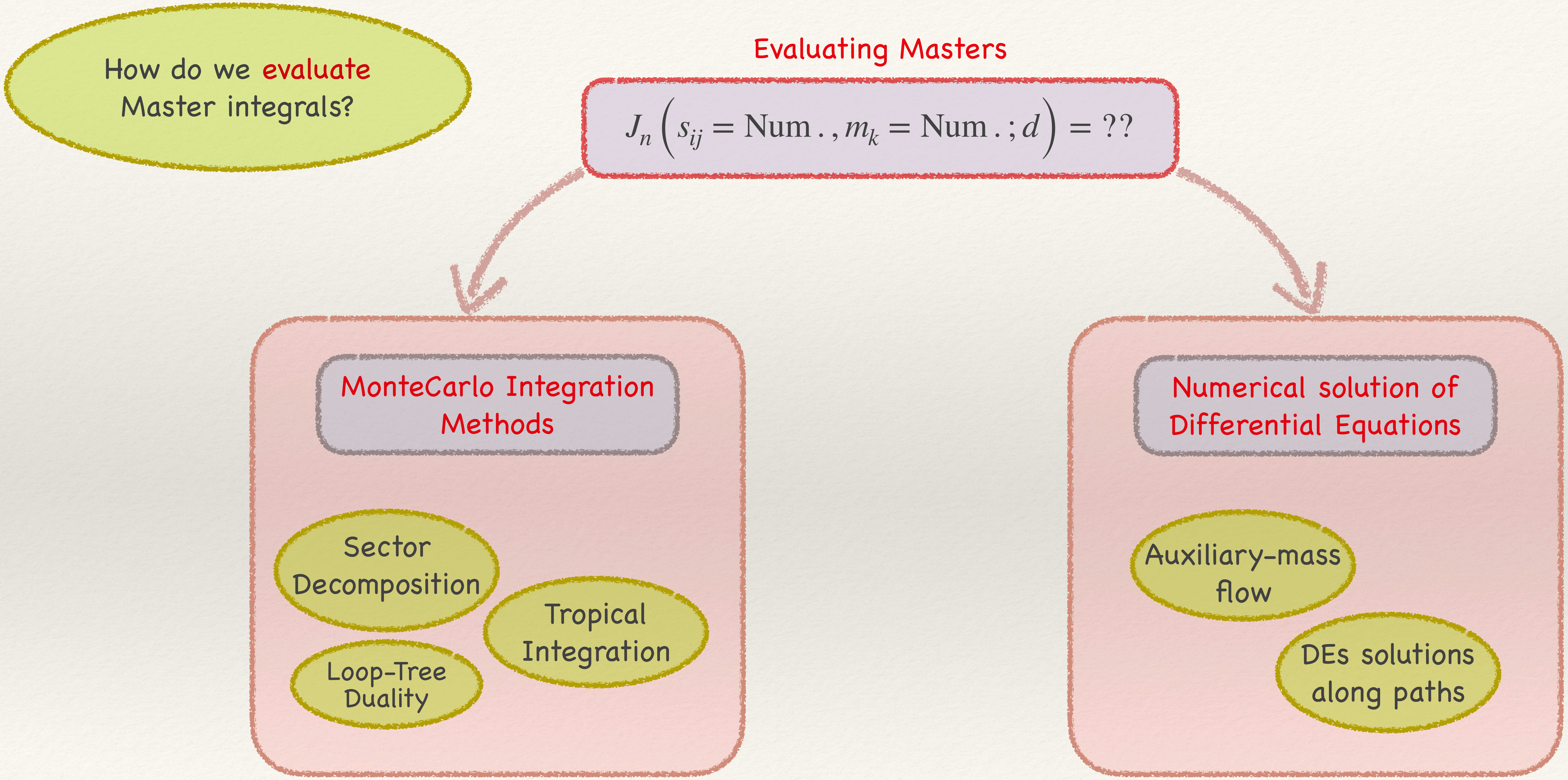


Reduced Interference

$$(\mathscr{M}_p^{n})_{ij} = \sum_{n} c_n \, J_n(\boldsymbol{s}, \boldsymbol{m}^2; d)$$

MASTER INTEGRALS

Numerical Integration



Numerical Integration

Auxiliary mass flow
(AMFlow)
[Liu,Ma:2201.11669]

- Introducing a mass parameter η into propagators
- Numerical IBPs + DE system depending on η only
- Automatic Boundary condition at $\eta \to \infty$
- Propagating boundaries to $\eta \to 0$

Series expansion methods
(DiffExp, SeaSyde, LINE)
[Hidding:2006.05510]
[Armadillo,Bonciani,Devoto,Rana,Vicini:2205.03345]
[Prisco,JR,Tramontano:to appear]

- Analytical IBPs + Differential Equation system
- Boundary condition as input
- Propagating boundary to input PS-points

Sector Decomposition
(SecDec, pySecDec)
[Heinrich,Jones,Kerner,Magerya,Olsson,Schlenk:2305.19768]

- Feynman parametrization
- Splitting integration domain
- End-point subtraction of singularities and ϵ expansion
- contour deformation + expansion-by-region
- MonteCarlo integration of finite integrals

Tropical integration
(FeynTrop)
[Borinski,Munch,Tellander:2302.08955]

- Feynman parameters + contour deformation $x_i \to x_i e^{-i\lambda \frac{d}{dx_i}\left(\frac{\mathcal{U}(\bar{x})}{\mathcal{F}(\bar{x})}\right)}$
- Tropical approximation of Symanzik Polynomial
- MonteCarlo integration improved with tropical sampling
- Improving sampling by geometrical insights

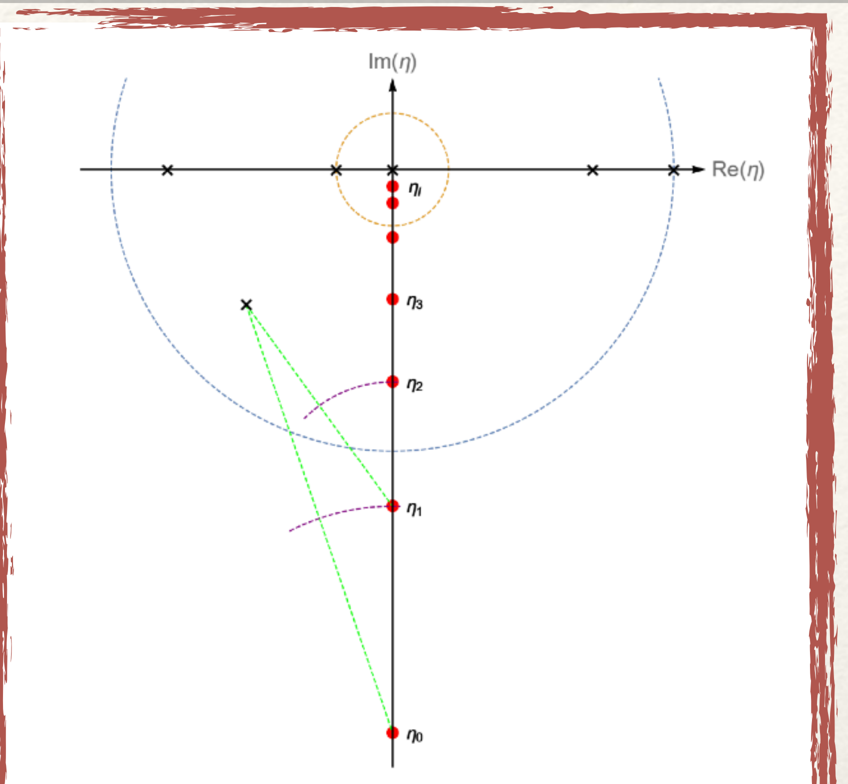


Figure 1: Singularities and analytic continuations. Singularities are labeled as crosses. Solid dots are points where to perform series expansions.

[Liu,Ma:2201.11669]

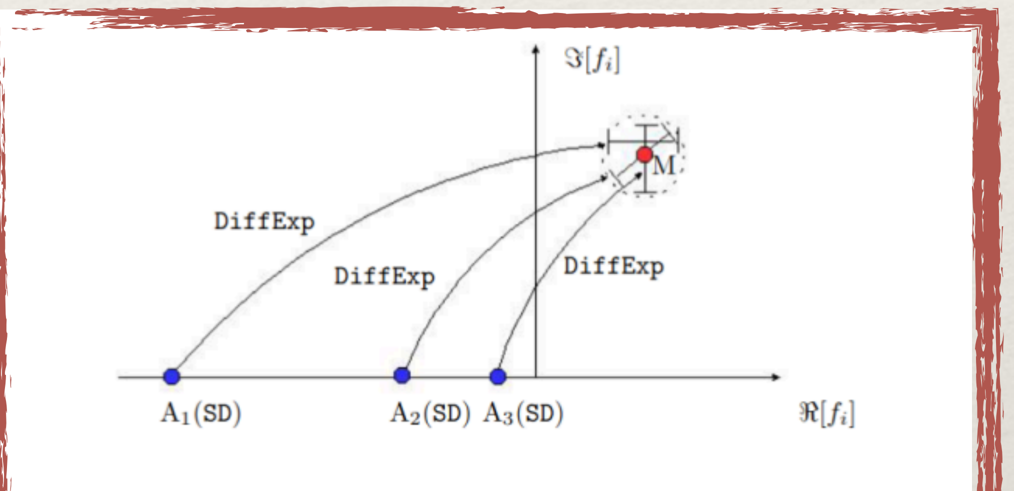
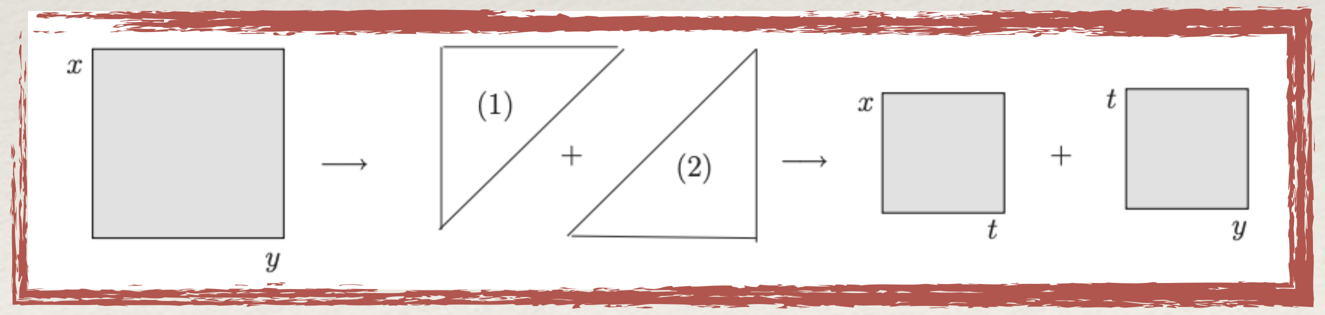


FIG. 1. Illustration of the DE transport method. The bound-

[Dubovyk,Freitas,Gluza,Grzanka,Hidding,Usovitsch:2201.0257]



[Heinrich:0803.4177]

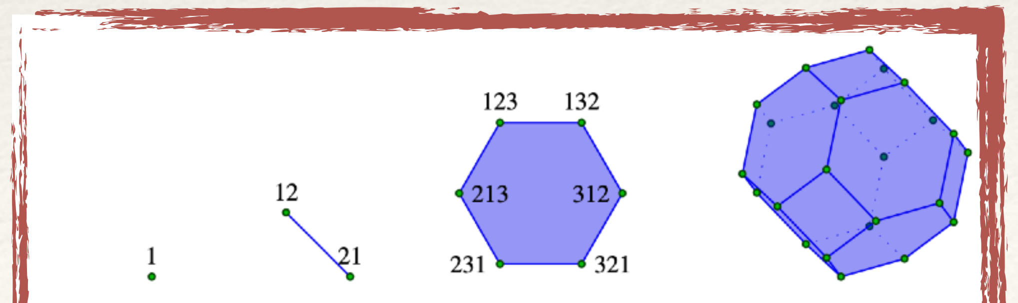


Figure 1: The first four permutohedra where it is easily seen from the explicit vertex coordinates (permutations) that all edges are parallel to $\mathbf{e}_i - \mathbf{e}_j$ for $i \neq j$.

[Borinsky,Munch,Tellander:2310.19890]

Codes references

QGraph
- [Nogueira:1991]

FeynArts
- [Küblbeck,Böhm,Denner:1991]
- [Hahn:hep-ph/0012260]

FeynCalc
- [Mertig,Böhm,Denner:1991]
- [Shtabovenko,Mertig,Orellana:2312.14089]

FORM
- [Vermaseren:math-ph/0010025]

Kira
- [Maierhöfer,Usovitsch,Uwer:1705.05610]
- [Klappert,Lange,Maierhöfer,Usovitsch:2008.06494]

LiteRED
- [Lee:1212.2685]

FIRE
- [Smirnov:2311.02370]

FiniteFlow
- [Peraro:1905.08019]

Blade
- [Guan,Liu,Ma,Wu:2405.14621]

pySecDec
- [Carter,Heinrich:1011.5493]
- [Heinrich,Jones,Kerner,Magerya,Olsson,Schlenk:2305.19768]

FeynTrop
- [Borinski,Munch,Tellander:2302.08955]

AMFlow
- [Liu,Ma:2201.11669]

DiffExp
- [Hidding:2006.05510]

SeaSyde
- [Armadillo,Bonciani,Devoto,Rana,Vicini:2205.03345]

LINE
- [Prisco,JR,Tramontano:to appear]

Summary

We present a novel code for automated evaluation of Scattering Amplitudes

LoopIn: **Loop In**tegrals

Features:

- Automated framework for evaluation of scattering amplitudes in pQFT++
- Designed for parallelization
- Modular structure, easily upgradable
- Tested on many 1L and 2L virtual correction in QED/QCD

Ongoing applications:

- tackling 3L QED processes
- Incoming updates: FORM + LINE

Long term:

- Robust grouping implementation
- Form factors and helicity amplitudes

LOOPIN

LOOPIN

Thank you for your attention!