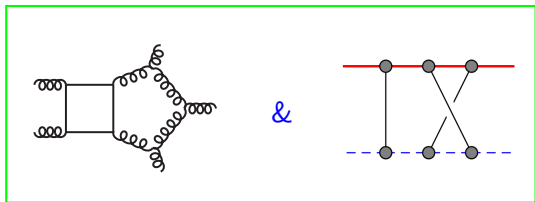


# Numerical Techniques for Scattering Amplitudes

## The Multi-Loop Numerical Unitarity Method



Fernando Febres Cordero

Department of Physics, Florida State University

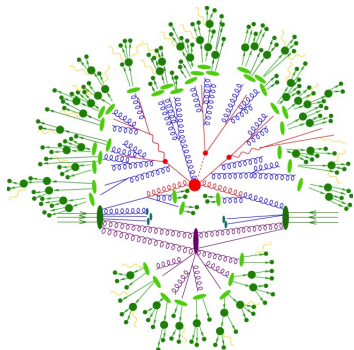
New Trends in QFT, Amplitudes and Gravity,  
CIIEC, BUAP, Mexico, April 2026

## Modern Techniques in Scattering Amplitudes

- Elvang, Huang; [Scattering Amplitudes](#), [[arXiv:1308.1697](#)]
- Weinzierl; [Feynman Integrals](#), [[arXiv:2201.03593](#)]
- Abreu, Duhr, Britto; [Mathematical structures in Feynman integrals](#), [[arXiv:2203.13014](#)]
- [Amplitudes 2023 Summer School](#), [Indico](#)

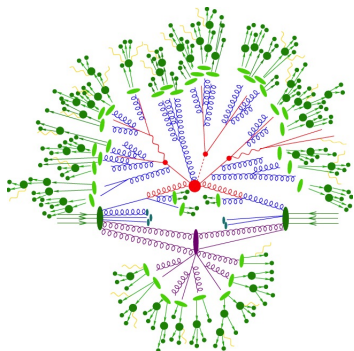
# Factorization for Hadron Collider Phenomenology

# Factorization for Hadron Collider Phenomenology



- Factorization
- Hard scattering
- Parton evolution
- Simulation underlying event and hadronization
- Particle decays and radiation

# Factorization for Hadron Collider Phenomenology



- Factorization
- Hard scattering
- Parton evolution
- Simulation underlying event and hadronization
- Particle decays and radiation

$$\sigma_{h_1 h_2 \rightarrow H} = \sum_{a,b} \int dx_a dx_b f_{a/h_1}(x_a, \mu_F) f_{b/h_2}(x_b, \mu_F) \hat{\sigma}_{ab \rightarrow H+X}(\mu_F, \mu_R)$$

# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^\kappa \left( \sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots \right)$$

# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^k (\sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots)$$

LO:



$\alpha_s^k$



# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^k (\sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots)$$

LO:

$\sim$

$\alpha_s^k$



NLO:

+

$\alpha_s^{k+1}$



# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^k (\sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots)$$

LO:

$\sim$

$\alpha_s^k$



NLO:

+

$\alpha_s^{k+1}$



NNLO:

+

$\alpha_s^{k+2}$



# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^k (\sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots)$$

LO:

$\sim$

$\alpha_s^k$



NLO:

+

$\alpha_s^{k+1}$



NNLO:

+

$\alpha_s^{k+2}$



N<sup>3</sup>LO:

+

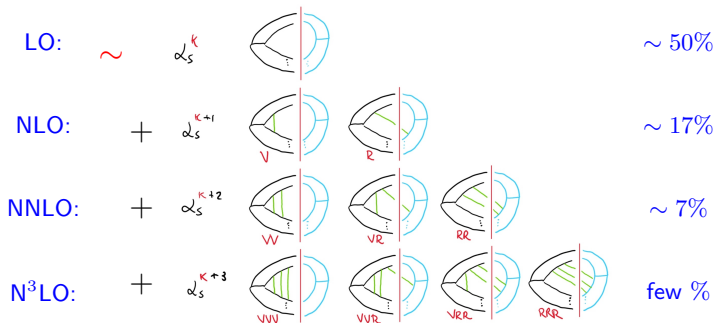
$\alpha_s^{k+3}$



# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^k (\sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots)$$

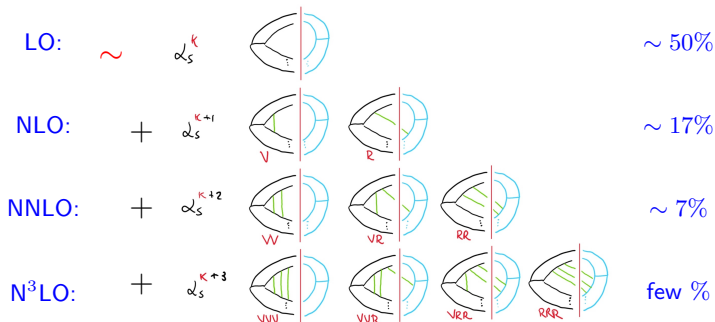
typical th uncertainties



# Truncation and Underlying Amplitudes

$$\hat{\sigma}_{ab \rightarrow H+X} = \alpha_s^k (\sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots)$$

typical th uncertainties



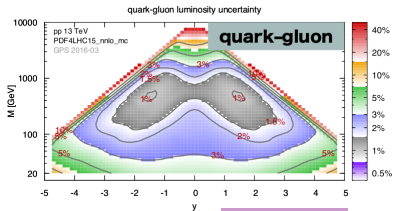
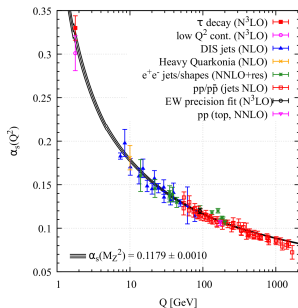
A **myriad** of amplitudes are required for precision calculations

# Uncertainties in Perturbative Predictions

## Perturbative truncation

$$\hat{\sigma}_{ab \rightarrow H} = \alpha_s^{\kappa} \left( \sigma_{\text{LO}} + \alpha_s \sigma_{\text{NLO}} + \alpha_s^2 \sigma_{\text{NNLO}} + \alpha_s^3 \sigma_{\text{N}^3\text{LO}} + \dots \right)$$

Parametric: determination of model's parameters  
couplings, PDFs, masses



From G. Salam

Amplitudes, IBP Identities, Numerical Unitarity

Exact Kinematics, Analytics from Numerics, Applications

# A Common Approach to Multi-Loop Amplitudes

## Feynman diagrams

↓  
Tensor reduction  
[Passarino, Veltman '79]

IBPs  
[Tkachov, Chetyrkin '81]



## Sum of master integrals

$$A = \sum_{\Gamma \in \Delta} \sum_{i \in M_{\Gamma}} c_{\Gamma,i} I_{\Gamma,i}$$

↓  
Differential equations

[Kotikov '91; Remiddi '97; Gehrmann, Remiddi '01; Henn '13]



Integrated form

# A Common Approach to Multi-Loop Amplitudes

## Feynman diagrams

|  
Tensor reduction  
[Passarino, Veltman '79]

IBPs  
[Tkachov, Chetyrkin '81]



## Sum of master integrals

$$A = \sum_{\Gamma \in \Delta} \sum_{i \in M_{\Gamma}} c_{\Gamma,i} I_{\Gamma,i}$$

|  
Differential equations

[Kotikov '91; Remiddi '97; Gehrmann, Remiddi '01; Henn '13]



Integrated form

General procedure, **but**:

- Large intermediate expressions
- Generating IBP relations is **practically difficult**
- Evaluating general **master integrals** efficiently still a major challenge

# A Common Approach to Multi-Loop Amplitudes

## Feynman diagrams

↓  
Tensor reduction  
[Passarino, Veltman '79]

↓  
IBPs  
[Tkachov, Chetyrkin '81]



## Sum of master integrals

$$A = \sum_{\Gamma \in \Delta} \sum_{i \in M_{\Gamma}} c_{\Gamma,i} I_{\Gamma,i}$$

↓  
Differential equations

[Kotikov '91; Remiddi '97; Gehrmann, Remiddi '01; Henn '13]



## Integrated form

General procedure, **but**:

- Large intermediate expressions
- Generating IBP relations is **practically difficult**
- Evaluating general **master integrals** efficiently still a major challenge

**Numerical** methods avoid some of these issues

- Perform reduction and evaluation **simultaneously**
- Avoid constructing an integrand: **numerical unitarity**
- Use the evaluations to 'lift' **analytic expressions**

## Multi-Loop Master Integrals

- Field theory gives us a representation of scattering amplitudes in terms of **Feynman diagrams**, where  $\mathcal{A} = \sum_i D_i$ .

# Multi-Loop Master Integrals

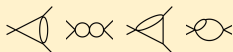
- Field theory gives us a representation of scattering amplitudes in terms of **Feynman diagrams**, where  $\mathcal{A} = \sum_i D_i$ .
- At a **given order in perturbation theory** this tells us:

$$\mathcal{A}^{(L)} = \int \left( \prod_{k=1}^L \frac{d^D \ell_k}{(2\pi)^D} \right) \sum_{\Gamma \in \Delta} \frac{\mathcal{N}_{\Gamma}(\ell_i; p_j; D)}{\prod_{k \in P_{\Gamma}} \rho_k}$$

# Multi-Loop Master Integrals

- Field theory gives us a representation of scattering amplitudes in terms of **Feynman diagrams**, where  $\mathcal{A} = \sum_i D_i$ .
- At a **given order in perturbation theory** this tells us:

$$\mathcal{A}^{(L)} = \int \left( \prod_{k=1}^L \frac{d^D \ell_k}{(2\pi)^D} \right) \sum_{\Gamma \in \Delta} \frac{\mathcal{N}_{\Gamma}(\ell_i; p_j; D)}{\prod_{k \in P_{\Gamma}} \rho_k}$$



The full diagram hierarchy  $\Delta$  for 2-loop 4-point planar amplitudes

# Multi-Loop Master Integrals

- Field theory gives us a representation of scattering amplitudes in terms of **Feynman diagrams**, where  $\mathcal{A} = \sum_i D_i$ .
- At a **given order in perturbation theory** this tells us:

$$\mathcal{A}^{(L)} = \int \left( \prod_{k=1}^L \frac{d^D \ell_k}{(2\pi)^D} \right) \sum_{\Gamma \in \Delta} \frac{\mathcal{N}_{\Gamma}(\ell_i; p_j; D)}{\prod_{k \in P_{\Gamma}} \rho_k}$$

# Multi-Loop Master Integrals

- Field theory gives us a representation of scattering amplitudes in terms of **Feynman diagrams**, where  $\mathcal{A} = \sum_i D_i$ .
- At a **given order in perturbation theory** this tells us:

$$\mathcal{A}^{(L)} = \int \left( \prod_{k=1}^L \frac{d^D \ell_k}{(2\pi)^D} \right) \sum_{\Gamma \in \Delta} \frac{\mathcal{N}_{\Gamma}(\ell_i; p_j; D)}{\prod_{k \in P_{\Gamma}} \rho_k}$$

- It **has been proven** that at a given loop order there is a **process-independent** and **finite basis** of functions that describe all Feynman integrals. These are the so-called **Master Integrals**

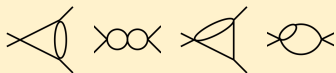
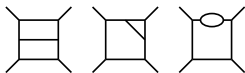
# Multi-Loop Master Integrals

- Field theory gives us a representation of scattering amplitudes in terms of **Feynman diagrams**, where  $\mathcal{A} = \sum_i D_i$ .
- At a **given order in perturbation theory** this tells us:

$$\mathcal{A}^{(L)} = \int \left( \prod_{k=1}^L \frac{d^D \ell_k}{(2\pi)^D} \right) \sum_{\Gamma \in \Delta} \frac{\mathcal{N}_{\Gamma}(\ell_i; p_j; D)}{\prod_{k \in P_{\Gamma}} \rho_k}$$

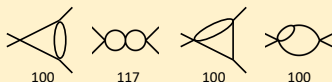
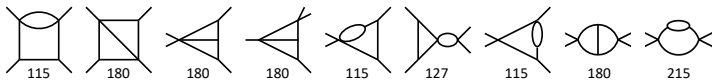
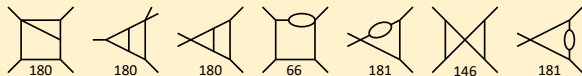
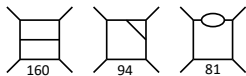
- It **has been proven** that at a given loop order there is a **process-independent** and **finite basis** of functions that describe all Feynman integrals. These are the so-called **Master Integrals**
- Proofs were provided by **Smirnov and Petukhov** [[arXiv:1004.4199](#)] and by **Bitoun, Bogner, Klausen, Panzer** [[arXiv:1712.09215](#)]

# The Planar Two-Loop Four-Point Hierarchy



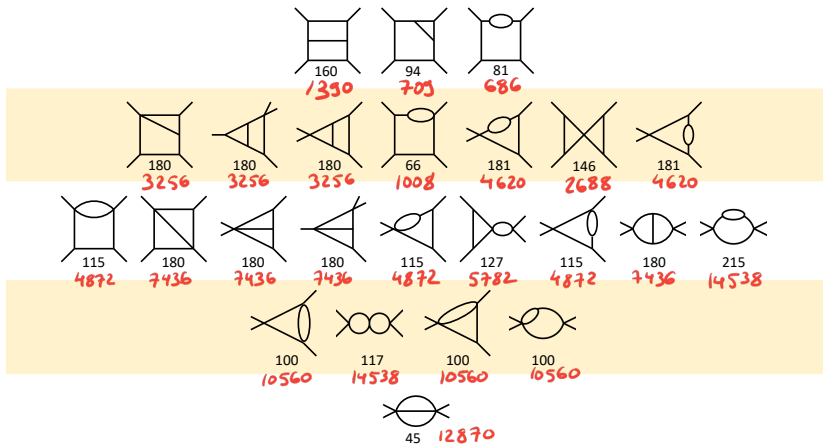
The full diagram hierarchy  $\Delta$  for 2-loop 4-point amplitudes

# The Planar Two-Loop Four-Point Hierarchy



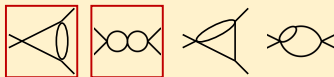
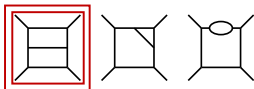
Number of independent integrals for a **renormalizable theory**

# The Planar Two-Loop Four-Point Hierarchy



Number of independent integrals in Einstein-Hilbert gravity

# The Planar Two-Loop Four-Point Hierarchy



The **propagator structures** with **master integrals** (& multiplicity)

# Linear Relations among Feynman Integrals

$$I(\vec{x}; \nu; D) = \int \left( \prod_{i=1}^L e^{\gamma_i \varepsilon} \frac{d^D k_i}{i \pi^{D/2}} \right) \frac{N(\{k_i \cdot k_j, b_i, p_j\}; D)}{\prod_{j=1}^P (m_j^2 - q_j^2 - i\varepsilon)^{\nu_j}}$$

kinematics      Prop power      dimension  
 $D = D_0 - 2\varepsilon$

→ for fixed  $\vec{x}$  &  $D \rightarrow$  discrete family of integrals with  $\nu \in \mathbb{Z}^P$

# Linear Relations among Feynman Integrals

$$I(\vec{x}; \nu; D) = \int \left( \frac{L}{i} e^{y \varepsilon} \frac{d^D k_i}{i \pi^{D/2}} \right) \frac{N(\{k_i, k_j, b_i, p_j, z_i, D\})}{\prod_{j=1}^P (m_j^2 - q_j^2 - i \varepsilon)^{\nu_j}}$$

kinematica    Prop power    dimension  
 $D = D_0 - 2\varepsilon$

→ for fixed  $\vec{x}$  &  $D \rightarrow$  discrete family of integrals with  $\nu \in \mathbb{Z}^P$

Proposition: there exist linear recursion relations for integrals with different exponents  $\nu \in \mathbb{Z}^P$ . Coefficients of relations are rational functions in  $\vec{x}$  &  $D$

# Finding Linear Relations

Write  $I(\vec{x}, \nu; \mathbb{D}) = \int \left( \prod_{j=1}^L e^{i x_j \nu_j} \right) F_I$  .

$\uparrow$   
the integrand

Then: Proposition:  $\int \prod_{j=1}^L d^D k_j \frac{\partial}{\partial k_j^\mu} \left[ \gamma^\mu F_I \right] = 0 \quad j=1,2,\dots,L$

$\uparrow$   
any vector

# Finding Linear Relations

Write  $I(\vec{x}, \nu; D) = \int \left( \prod_{j=1}^L e^{x_j} \frac{d^D k_j}{i\pi^{D/2}} \right) F_I$  .

↑  
the integrand

Then: *Proposition:*  $\int d^D k_j \frac{\partial}{\partial k_j^\mu} \left[ \tilde{\nu}^\mu F_I \right] = 0 \quad j=1,2,\dots,L$

↑  
any vector

*Proof:*  $k_j^\mu \rightarrow \lambda k_j^\mu + \tilde{\nu}^\mu$  for  $\lambda \in \mathbb{R}^+$  &  $\tilde{\nu}^\mu$  a constant  $D$ -vector

$$\Rightarrow I = \lambda^D \int \left( \prod_{j=1}^L e^{x_j} \frac{d^D k_j}{i\pi^{D/2}} \right) F_I(k_j \rightarrow \lambda k_j^\mu + \tilde{\nu}^\mu)$$

# Finding Linear Relations

Write  $I(\vec{x}, \nu; D) = \int \left( \prod_{j=1}^L e^{i x_j} \frac{d^D k_j}{i \pi^{D/2}} \right) F_I$

↑  
the integrand

Then: *Proposition:*  $\int d^D k_j \frac{\partial}{\partial k_j^\mu} \left[ \tilde{\nu}^\mu F_I \right] = 0 \quad j=1, 2, \dots, L$

↑  
any vector

*Proof:*  $k_j^\mu \rightarrow \lambda k_j^\mu + \tilde{\nu}^\mu$  for  $\lambda \in \mathbb{R}^+$  &  $\tilde{\nu}^\mu$  a constant  $D$ -vector

$$\Rightarrow I = \lambda^D \int \left( \prod_{j=1}^L e^{i x_j} \frac{d^D k_j}{i \pi^{D/2}} \right) F_I(k_j \rightarrow \lambda k_j^\mu + \tilde{\nu}^\mu)$$

Make it infinitesimal:  $k \rightarrow k + \varepsilon \tilde{\nu}$

$$\Rightarrow I = I + \varepsilon \tilde{\nu}^\mu \int d^D k_j \frac{\partial}{\partial k_j^\mu} (F_I) + \mathcal{O}(\varepsilon^2)$$

# Finding Linear Relations

Write  $I(\vec{x}_3, \nu; D) = \int \left( \frac{L}{\pi} e^{e^{\chi} \frac{d^D k_j}{i\pi D \epsilon}} \right) F_I$

↑  
the integrand

Then: Proposition:  $\int d^D k_j \frac{\partial}{\partial k_j^M} \left[ \tilde{\nu}^M F_I \right] = 0 \quad j=1,2,\dots,L$

↑  
any vector

Proof:  $k_j^M \rightarrow \lambda k_j^M + \tilde{\nu}^M$  for  $\lambda \in \mathbb{R}^+$  &  $\tilde{\nu}^M$  a constant  $D$ -vector

$$\Rightarrow I = \lambda^D \int \left( \frac{L}{\pi} e^{e^{\chi} \frac{d^D k_j}{i\pi D \epsilon}} \right) F_I(k_j \rightarrow \lambda k_j^M + \tilde{\nu}^M)$$

Make it infinitesimal:  $k \rightarrow k + \epsilon \tilde{\nu}$

$$\Rightarrow I = I + \epsilon \tilde{\nu}^M \int d^D k_j \frac{\partial}{\partial k_j^M} (F_I) + \mathcal{O}(\epsilon^2)$$

$$\Rightarrow \boxed{\int d^D k_j \frac{\partial}{\partial k_j^M} (\tilde{\nu}^M F_I) = 0}$$

# Finding Linear Relations

Write  $I(\vec{x}_3, \nu; D) = \int \left( \prod_{j=1}^L e^{e^{\nu_j} \frac{d^D k_j}{i\pi^{D/2}}} \right) F_I$

↑  
the integrand

Then: Proposition:  $\int d^D k_j \frac{\partial}{\partial k_j^\mu} \left[ \tilde{\nu}^\mu F_I \right] = 0 \quad j=1, 2, \dots, L$

↑  
any vector

Proof:  $k_j^\mu \rightarrow \lambda k_j^\mu + \tilde{\nu}^\mu$  for  $\lambda \in \mathbb{R}^+$  &  $\tilde{\nu}^\mu$  a constant  $D$ -vector

$$\Rightarrow I = \lambda^D \int \left( \prod_{j=1}^L e^{e^{\nu_j} \frac{d^D k_j}{i\pi^{D/2}}} \right) F_I(k_j \rightarrow \lambda k_j^\mu + \tilde{\nu}^\mu)$$

Make it infinitesimal:  $k \rightarrow k + \varepsilon \tilde{\nu}$

$$\Rightarrow I = I + \varepsilon \tilde{\nu}^\mu \int d^D k_j \frac{\partial}{\partial k_j^\mu} (F_I) + \mathcal{O}(\varepsilon^2)$$

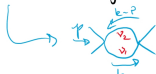
$$\Rightarrow \boxed{\int d^D k_j \frac{\partial}{\partial k_j^\mu} (\tilde{\nu}^\mu F_I) = 0}$$

Similarly:  $k \rightarrow e^\varepsilon k \Rightarrow \boxed{\int d^D k_j \frac{\partial}{\partial k_j^\mu} (k_j^\mu F_I) = 0}$

QED.

# IBP Identities: An Example

EXAMPLE:  $I(v_1, v_2) = e^{\delta \epsilon} \int_0^1 \frac{dk}{i\pi^{D/2}} \frac{1}{(k^2)^{\nu_1} (k-p)^{\nu_2}}$



IBP-1:  $\mathcal{U}^M = k^M$

$$\partial_\mu \left( \frac{k^\mu}{(i)^{\nu_1} (i)^{\nu_2}} \right) = D \frac{1}{(i)^{\nu_1} (i)^{\nu_2}} + 2\nu_1 \frac{k^2}{(i)^{\nu_1} (i)^{\nu_2}} + 2\nu_2 \frac{k \cdot p}{(i)^{\nu_1} (i)^{\nu_2+1}}$$

$$\Lambda_2 \int_0^1 dk \partial_\mu \left( \frac{k^\mu}{(i)^{\nu_1} (i)^{\nu_2}} \right) = 0 \text{ then:}$$

$$0 = D I(v_1, v_2) - 2\nu_1 I(v_1, v_2) - \nu_2 I(v_1, v_2) - \nu_2 I(v_1-1, v_2+1) - \nu_2 p^2 I(v_1, v_2+1)$$



IBP-2:  $\mathcal{U}^M = p^M$

$$\Rightarrow 0 = (v_1 - v_2) I(v_1, v_2) - v_1 I(v_1+1, v_2-1) - v_1 p^2 I(v_1+1, v_2) + v_2 I(v_1-1, v_2+1) + v_2 p^2 I(v_1, v_2+1)$$



# IBP Identities: An Example

By making the variable replacements:

$$\text{IBP-1: } \begin{aligned} \tilde{v}_1 &= v_1 \\ \tilde{v}_2 &= v_2 + 1 \end{aligned}$$

$$\text{IBP-2 (IBP-1): } \begin{aligned} \tilde{v}_1 &= v_1 + 1 \\ \tilde{v}_2 &= v_2 \end{aligned}$$

$$\textcircled{R1} \quad I(v_1, v_2) = \frac{D+1-2v_1-v_2}{p^2(v_2-1)} I(v_1, v_2-1) - \frac{1}{p^2} I(v_1-1, v_2)$$

$v_2 \neq 1$   
 $(v_1, v_2)$

$$\textcircled{R2} \quad I(v_1, v_2) = \frac{D+1-2v_2-v_1}{p^2(v_1-1)} I(v_1-1, v_2) - \frac{1}{p^2} I(v_1, v_2-1)$$

$v_1 \neq 1$   
 $(v_1, v_2)$



# IBP Identities: An Example

By making the variable replacements:

$$\text{IBP-1: } \begin{aligned} \tilde{v}_1 &= v_1 \\ \tilde{v}_2 &= v_2 + 1 \end{aligned}$$

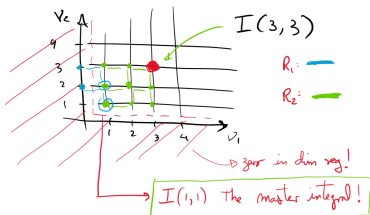
$$\text{IBP-2 (IBP-1): } \begin{aligned} \tilde{v}_1 &= v_1 + 1 \\ \tilde{v}_2 &= v_2 \end{aligned}$$

$$\textcircled{R_1} \quad I(v_1, v_2) = \frac{D+1-2v_1-v_2}{p^2(v_2-1)} I(v_1, v_2-1) - \frac{1}{p^2} I(v_1-1, v_2)$$

$v_2 \pm 1$

$$\textcircled{R_2} \quad I(v_1, v_2) = \frac{D+1-2v_2-v_1}{p^2(v_1-1)} I(v_1-1, v_2) - \frac{1}{p^2} I(v_1, v_2-1)$$

$v_1 \pm 1$



More complex integrals in  $\mathbb{Z}^p$  produce very intricate relations. Laporta<sup>(2001)</sup> constructed a partial order in  $\mathbb{Z}^p$  which gives an algorithm that helps in mapping generic integrals into master integrals!

Software  $\rightarrow$  KIRA (Freiburg)  
FIRE  
Reduze  
...



## The Numerical Unitarity Method

# Numerical Unitarity

Decompose  $\mathcal{A}$  in terms of *master integrals*:

$$\mathcal{A}^{(L)} = \sum_{\Gamma \in \Delta} \sum_{i \in M_{\Gamma}} c_{\Gamma,i} \mathcal{I}_{\Gamma,i}$$

$\{\mathcal{I}_{\Gamma,i}\}$  is **process independent** and **finite**

# Numerical Unitarity

Decompose  $\mathcal{A}$  in terms of *master integrals*:

$$\mathcal{A}^{(L)} = \sum_{\Gamma \in \Delta} \sum_{i \in M_{\Gamma}} c_{\Gamma,i} \mathcal{I}_{\Gamma,i}$$

$\{\mathcal{I}_{\Gamma,i}\}$  is **process independent** and **finite**

Drop the integral symbol, introducing the **integrand ansatz**:

$$\mathcal{A}^{(L)}(\ell_l) = \sum_{\Gamma \in \Delta} \sum_{k \in Q_{\Gamma}} c_{\Gamma,k} \frac{m_{\Gamma,k}(\ell_l)}{\prod_{j \in P_{\Gamma}} \rho_j(\ell_l)}$$

Functions  $Q_{\Gamma} = \{m_{\Gamma,k}(\ell_l) | k \in Q_{\Gamma}\}$  **parametrize** every possible integrand (up to a given power of loop momenta). **E.g.:**

# Numerical Unitarity

Decompose  $\mathcal{A}$  in terms of *master integrals*:

$$\mathcal{A}^{(L)} = \sum_{\Gamma \in \Delta} \sum_{i \in M_{\Gamma}} c_{\Gamma,i} \mathcal{I}_{\Gamma,i}$$

$\{\mathcal{I}_{\Gamma,i}\}$  is **process independent** and **finite**

Drop the integral symbol, introducing the **integrand ansatz**:

$$\mathcal{A}^{(L)}(\ell_l) = \sum_{\Gamma \in \Delta} \sum_{k \in Q_{\Gamma}} c_{\Gamma,k} \frac{m_{\Gamma,k}(\ell_l)}{\prod_{j \in P_{\Gamma}} \rho_j(\ell_l)}$$

Functions  $Q_{\Gamma} = \{m_{\Gamma,k}(\ell_l) | k \in Q_{\Gamma}\}$  **parametrize** every possible integrand (up to a given power of loop momenta). **E.g.:**

- **Tensor Basis**
- **Scattering Plane Tensor Basis**
- **Master-Surface Basis**

- Given diagram  $\Gamma$  make an *adaptive momentum parametrization*

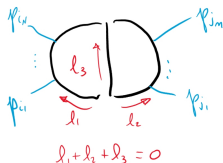
(related [ven Neerven, Vermaseren], [Ossola, Papadopoulos, Pittau], [Bobadilla, Mastroliia, Peraro, Primo])

$$l_l = \sum_{j \in B_l^p} v_l^j r^{lj} + \underbrace{\sum_{j \in B_l^t} v_l^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common Transverse}} + \sum_{i \in B^e} n^i \mu_l^i$$

- Given diagram  $\Gamma$  make an *adaptive momentum parametrization*

(related [ven Neerven, Vermaseren], [Ossola, Papadopoulos, Pittau], [Bobadilla, Mastrolia, Peraro, Primo])

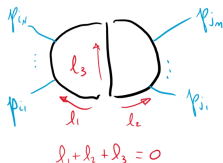
$$l_i = \sum_{j \in B_i^p} v_i^j r^{lj} + \underbrace{\sum_{j \in B_i^t} v_i^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common Transverse}} + \sum_{i \in B^e} n^i \mu_i^i$$



- Given diagram  $\Gamma$  make an *adaptive momentum parametrization*

(related [ven Neerven, Vermaseren], [Ossola, Papadopoulos, Pittau], [Bobadilla, Mastroliia, Peraro, Primo])

$$l_i = \sum_{j \in B_i^p} v_i^j r^{lj} + \underbrace{\sum_{j \in B_i^t} v_i^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common transverse}} + \sum_{i \in B^e} n^i \mu_i^i$$

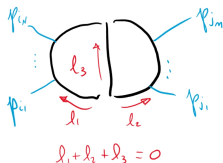


\*  $\{p_k\}$  external momenta defines a transverse 4-dim subspace generated by  $\{n_i; i \in B^{ct}\}$  such that  $p_k \cdot n_i = 0 \quad \forall k \neq i \in B^{ct}$

- Given diagram  $\Gamma$  make an *adaptive momentum parametrization*

(related [ven Neerven, Vermaseren], [Ossola, Papadopoulos, Pittau], [Bobadilla, Mastrolia, Peraro, Primo])

$$l_i = \sum_{j \in B_i^p} v_i^j r^{lj} + \underbrace{\sum_{j \in B_i^t} v_i^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common transverse}} + \sum_{i \in B^e} n^i \mu_i^i$$



\*  $\{p_k\}$  external momenta defines a transverse 4-dim subspace generated by  $\{n_i; i \in B^{ct}\}$  such that

$$p_k \cdot n_i = 0 \quad \forall k \neq i \in B^{ct}$$

\* Consider  $l_e$   $\rightarrow$  complete 4-dim space with

$$\text{Gen}\{p_{e_i}; \forall i\} \cup \text{Gen}\{n_i; i \in B^{ct}\}$$

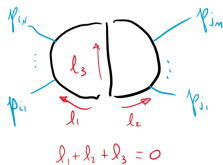
we do that with  $\{v_i; i \in B^t\}$  such that

$v_i$  is dual to  $p_{j_k}$  with  $j \neq l$

- Given diagram  $\Gamma$  make an *adaptive momentum parametrization*

(related [ven Neerven, Vermaseren], [Ossola, Papadopoulos, Pittau], [Bobadilla, Mastrolia, Peraro, Primo])

$$l_i = \sum_{j \in B_i^p} v_i^j r^{lj} + \underbrace{\sum_{j \in B_i^t} v_i^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common transverse}} + \sum_{i \in B^e} n^i \mu_i^i$$

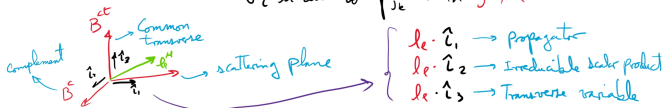


\*  $\{p_k\}$  external momenta defines a transverse 4-dim subspace generated by  $\{n_i; i \in B^{ct}\}$  such that

$$p_k \cdot n_i = 0 \quad \forall k \neq i \in B^{ct}$$

\* Consider  $\{l_e\} \rightarrow$  complete 4-dim space with  $\text{Gen}\{p_{e_i}; \forall i\} \cup \text{Gen}\{n_i; i \in B^{ct}\}$

we do that with  $\{v_i; i \in B^t\}$  such that  $v_i$  is dual to  $p_{j_k}$  with  $j \neq l$



# Integrand Parametrizations

- Exploit the variables of the **adaptive momentum parametrization** to build **integrand parametrizations**

$$\ell_l = \sum_{j \in B_l^p} v_l^j r^{lj} + \underbrace{\sum_{j \in B_l^i} v_l^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common Transverse}} + \sum_{i \in B^e} n^i \mu_l^i$$

# Integrand Parametrizations

- Exploit the variables of the **adaptive momentum parametrization** to build **integrand parametrizations**

$$\ell_l = \sum_{j \in B_l^p} v_l^j r^{lj} + \underbrace{\sum_{j \in B_l^t} v_l^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common Transverse}} + \sum_{i \in B^e} n^i \mu^i$$

- **Tensor Basis:** constructed from all **monomials**  $(\alpha^{lj})^{\vec{a}} (\alpha^{li})^{\vec{b}}$  with  $j \in B_l^t$  and  $i \in B^{ct}$

# Integrand Parametrizations

- Exploit the variables of the **adaptive momentum parametrization** to build **integrand parametrizations**

$$\ell_i = \sum_{j \in B_l^p} v_l^j r^{lj} + \underbrace{\sum_{j \in B_l^t} v_l^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common Transverse}} + \sum_{i \in B^e} n^i \mu_l^i$$

- **Tensor Basis:** constructed from all **monomials**  $(\alpha^{lj})^{\vec{a}} (\alpha^{li})^{\vec{b}}$  with  $j \in B_l^t$  and  $i \in B^{ct}$
- **Scattering Plane Tensor Basis:** constructed from all **monomials**  $(\alpha^{lj})^{\vec{a}}$  with  $j \in B_l^t$  and one-loop-like surface terms with **common transverse** variables [Abreu, FFC, Ita, Jaquier, Page, Zeng [arXiv:1703.05273]; see also [Bobadilla, Mastrolia, Peraro, Primo]

# Integrand Parametrizations

- Exploit the variables of the **adaptive momentum parametrization** to build **integrand parametrizations**

$$\ell_i = \sum_{j \in B_l^t} v_l^j r^{lj} + \underbrace{\sum_{j \in B_l^t} v_l^j \alpha^{lj}}_{\text{ISP's}} + \underbrace{\sum_{i \in B^{ct}} \frac{n^i}{(n^i)^2} \alpha^{li}}_{\text{Common Transverse}} + \sum_{i \in B^e} n^i \mu_l^i$$

- **Tensor Basis:** constructed from all **monomials**  $(\alpha^{lj})^{\vec{a}} (\alpha^{li})^{\vec{b}}$  with  $j \in B_l^t$  and  $i \in B^{ct}$
- **Scattering Plane Tensor Basis:** constructed from all **monomials**  $(\alpha^{lj})^{\vec{a}}$  with  $j \in B_l^t$  and one-loop-like surface terms with **common transverse** variables [Abreu, FFC, Ita, Jaquier, Page, Zeng [arXiv:1703.05273]; see also [Bobadilla, Mastrolia, Peraro, Primo]
- **Master-Surface Basis:** powerful parametrization trivializing **map to master integrals** [Ita, arXiv:1510.05626]. That is,  $Q_\Gamma = M_\Gamma \cup S_\Gamma$ :

$$\int \frac{d^D \ell_1 d^D \ell_2}{(2\pi)^{2D}} \frac{m_{\Gamma,i}(\ell_l)}{\prod_{k \in P_\Gamma} \rho_k} = \begin{cases} I_{\Gamma,i} & \text{for } i \in M_\Gamma \quad (\text{master}) \\ 0 & \text{for } i \in S_\Gamma \quad (\text{surface}) \end{cases}$$

# Scattering Plane Tensor Basis

# Scattering Plane Tensor Basis

- Implicitly **defined** in early developments [[arXiv:1703.05273](#)], [[arXiv:1712.03946](#)], ...
- **Named and automated** in **CARAVEL** for generic 2-loop calculations in [Akpinar, FFC, Kraus, Zeng](#) [[arXiv:2407.19005](#)]
- Allows **straightforward map** to IBP programs
- **Key**: trivially **reduces transverse integrals** (no tensor decomposition in **computing helicity amplitudes!**)

# Scattering Plane Tensor Basis

- Implicitly **defined** in early developments [[arXiv:1703.05273](#)], [[arXiv:1712.03946](#)], ...
- **Named and automated** in **CARAVEL** for generic 2-loop calculations in [Akpinar, FFC, Kraus, Zeng](#) [[arXiv:2407.19005](#)]
- Allows **straightforward map** to IBP programs
- **Key**: trivially **reduces transverse integrals** (no tensor decomposition in **computing helicity amplitudes!**)

Typically less than  $\mathcal{O}(5\%)$  (IBP-)reducible monomials (with scattering-plane variables) remain

Consider the **integration by parts (IBP)** relation on  $\Gamma$

$$0 = \int \prod_i d^D \ell_i \frac{\partial}{\partial \ell_j^\nu} \left[ \frac{w_j^\nu}{\prod_{k \in P_\Gamma} \rho_k} \right]$$

making it *unitarity compatible* (controlling the **propagator structure**) [Gluza, Kadja, Kosower '10; Schabinger '11]

$$w_j^\nu \frac{\partial}{\partial \ell_j^\nu} \rho_k = f_k \rho_k$$

Write ansatz for  $w_j^\nu$  expanded in external and loop momenta, and solve polynomial equations using **algebraic geometry** techniques

Build a full set of surface terms and fill the rest of the space with **master integrands**

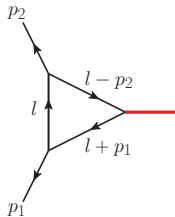
See recent (massive!) developments by Page and Song [arXiv:2509.17681]

and Coro, Novichkov, Page and Song [arXiv:2512.05869]

## A 1-loop Example for Surface Terms: Part 1

Consider the 1-loop 1-mass triangle with

$$\rho_1 = (\ell + p_1)^2, \quad \rho_2 = \ell^2, \quad \rho_3 = (\ell - p_2)^2$$



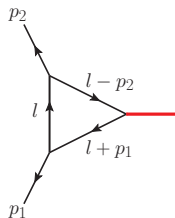
# A 1-loop Example for Surface Terms: Part 1

Consider the 1-loop 1-mass triangle with

$$\rho_1 = (\ell + p_1)^2, \quad \rho_2 = \ell^2, \quad \rho_3 = (\ell - p_2)^2$$

and we construct  $u^\nu \partial / \partial \ell^\nu$  by parametrizing

$$u^\nu = u_1^{\text{ext}} p_1^\nu + u_2^{\text{ext}} p_2^\nu + u^{\text{loop}} \ell^\nu$$



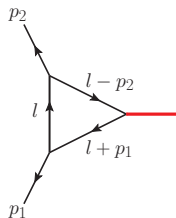
# A 1-loop Example for Surface Terms: Part 1

Consider the 1-loop 1-mass triangle with

$$\rho_1 = (\ell + p_1)^2, \quad \rho_2 = \ell^2, \quad \rho_3 = (\ell - p_2)^2$$

and we construct  $u^\nu \partial / \partial \ell^\nu$  by parametrizing

$$u^\nu = u_1^{\text{ext}} p_1^\nu + u_2^{\text{ext}} p_2^\nu + u^{\text{loop}} \ell^\nu$$



We then get the **syzygy equation** (polynomial equation):

$$(u_1^{\text{ext}} p_1^\nu + u_2^{\text{ext}} p_2^\nu + u^{\text{loop}} \ell^\nu) \frac{\partial}{\partial \ell^\nu} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} - \begin{pmatrix} f_1 \rho_1 \\ f_2 \rho_2 \\ f_3 \rho_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

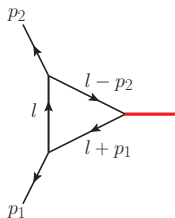
# A 1-loop Example for Surface Terms: Part 1

Consider the 1-loop 1-mass triangle with

$$\rho_1 = (\ell + p_1)^2, \quad \rho_2 = \ell^2, \quad \rho_3 = (\ell - p_2)^2$$

and we construct  $u^\nu \partial / \partial \ell^\nu$  by parametrizing

$$u^\nu = u_1^{\text{ext}} p_1^\nu + u_2^{\text{ext}} p_2^\nu + u^{\text{loop}} \ell^\nu$$



We then get the **syzygy equation** (polynomial equation):

$$(u_1^{\text{ext}} p_1^\nu + u_2^{\text{ext}} p_2^\nu + u^{\text{loop}} \ell^\nu) \frac{\partial}{\partial \ell^\nu} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} - \begin{pmatrix} f_1 \rho_1 \\ f_2 \rho_2 \\ f_3 \rho_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

We can then show that we have the solution for the **IBP-generating vector**:

$$u^\nu \frac{\partial}{\partial \ell^\nu} = [(\rho_3 - \rho_2) p_1^\nu + (\rho_1 + \rho_2) p_2^\nu + (-s + 2\rho_3 - 2\rho_2) \ell^\nu] \frac{\partial}{\partial \ell^\nu}$$

## A 1-loop Example for Surface Terms: Part 2

Now we have the surface term:

$$0 = \int d^D \ell \frac{\partial}{\partial \ell^\nu} \left( \frac{u^\nu}{\rho_1 \rho_2 \rho_3} \right) = \int d^D \ell \frac{1}{\rho_1 \rho_2 \rho_3} [-(D-4)s - 2(D-3)\rho_2 + 2(D-3)\rho_3]$$

## A 1-loop Example for Surface Terms: Part 2

Now we have the surface term:

$$0 = \int d^D \ell \frac{\partial}{\partial \ell^\nu} \left( \frac{u^\nu}{\rho_1 \rho_2 \rho_3} \right) = \int d^D \ell \frac{1}{\rho_1 \rho_2 \rho_3} [-(D-4)s - 2(D-3)\rho_2 + 2(D-3)\rho_3]$$

The scalar triangle integrand **can be replaced by a surface term**, though commonly it is kept in 1-loop calculations, calling it a “master” integral.

## A 1-loop Example for Surface Terms: Part 2

Now we have the surface term:

$$0 = \int d^D \ell \frac{\partial}{\partial l^\nu} \left( \frac{u^\nu}{\rho_1 \rho_2 \rho_3} \right) = \int d^D \ell \frac{1}{\rho_1 \rho_2 \rho_3} [-(D-4)s - 2(D-3)\rho_2 + 2(D-3)\rho_3]$$

The scalar triangle integrand **can be replaced by a surface term**, though commonly it is kept in 1-loop calculations, calling it a “master” integral.

Notice the **IBP identity** between the triangle and the  $s = (p_1 + p_2)^2$  bubble is:

$$-(D-4)sI_{\text{tri}} - 2(D-3)I_{\text{s-bub}} = 0$$

## A 1-loop Example for Surface Terms: Part 2

Now we have the surface term:

$$0 = \int d^D \ell \frac{\partial}{\partial \ell^\nu} \left( \frac{u^\nu}{\rho_1 \rho_2 \rho_3} \right) = \int d^D \ell \frac{1}{\rho_1 \rho_2 \rho_3} [-(D-4)s - 2(D-3)\rho_2 + 2(D-3)\rho_3]$$

The scalar triangle integrand **can be replaced by a surface term**, though commonly it is kept in 1-loop calculations, calling it a “master” integral.

Notice the **IBP identity** between the triangle and the  $s = (p_1 + p_2)^2$  bubble is:

$$-(D-4)sI_{\text{tri}} - 2(D-3)I_{s\text{-bub}} = 0$$

Similar manipulations can be carried out at **higher loops**. More complicated **syzygy equations** (polynomial relations) need to be solved (using e.g. **algebraic geometry** techniques)

Solutions to  $u_j^\nu$  are power-counting independent. When parametrizing a given numerator of a  $\Gamma \in \Delta$  we need to consider the required power-counting for the theory at hand.

But we can *industrially* produce surface terms by considering polynomials  $t_r(\ell_l)$ , and then considering the vector  $t_r(\ell_l)u_j^\nu$ :

$$m_{\Gamma,(r,s)} = u_j^\nu \frac{\partial t_r(\ell_l)}{\partial \ell_i^\nu} + t_r(\ell_l) \left( \frac{\partial u_j^\nu}{\partial \ell_i^\nu} - \sum_{k \in P_\Gamma} f_k^s \right)$$

Solutions to  $u_j^\nu$  are power-counting independent. When parametrizing a given numerator of a  $\Gamma \in \Delta$  we need to consider the required power-counting for the theory at hand.

But we can *industrially* produce surface terms by considering polynomials  $t_r(\ell_l)$ , and then considering the vector  $t_r(\ell_l)u_j^\nu$ :

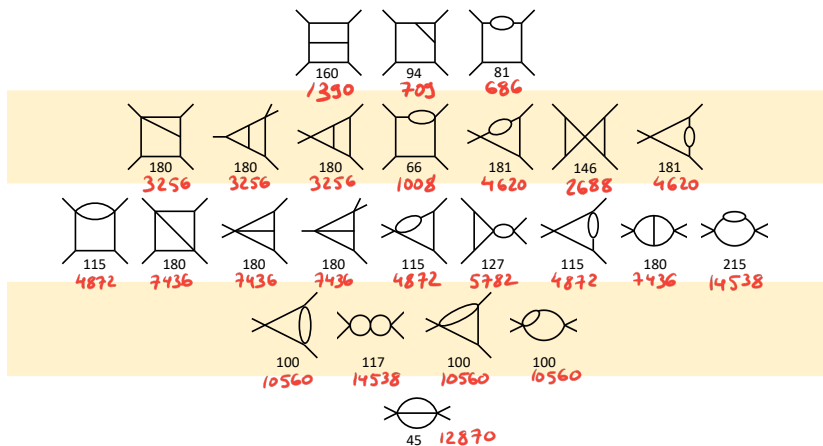
$$m_{\Gamma,(r,s)} = u_j^\nu \frac{\partial t_r(\ell_l)}{\partial \ell_i^\nu} + t_r(\ell_l) \left( \frac{\partial u_j^\nu}{\partial \ell_i^\nu} - \sum_{k \in P_\Gamma} f_k^s \right)$$

A four-graviton amplitude calculation in **Einstein gravity** structurally the same as a four-gluon amplitude calculation in **QCD**!

See recent (massive!) developments by Page and Song [arXiv:2509.17681]  
and Coro, Novichkov, Page and Song [arXiv:2512.05869]

see also similar approach in de la Cruz, Kosower [arXiv:2602.22111]

# The Planar Two-Loop Four-Point Hierarchy



Number of independent integrals in Einstein-Hilbert gravity

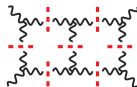
Compute the Integrand!

$$\mathcal{A}^{(L)}(\ell_l) = \sum_{\Gamma \in \Delta} \sum_{k \in Q_\Gamma} c_{\Gamma,k} \frac{m_{\Gamma,k}(\ell_l)}{\prod_{j \in P_\Gamma} \rho_j(\ell_l)}$$

# Unitarity Approach to Computing Integrand Coefficients

[Bern, Dixon, Dunbar, Kosower] [Britto, Cachazo, Feng]

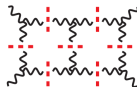
- In **on-shell configurations** of  $\ell_l$ , the integrand factorizes



# Unitarity Approach to Computing Integrand Coefficients

[Bern, Dixon, Dunbar, Kosower] [Britto, Cachazo, Feng]

- In **on-shell configurations** of  $\ell_l$ , the integrand factorizes and produces a **cut equation** (**partial fractioning** the integrand!)

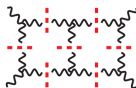


# Unitarity Approach to Computing Integrand Coefficients

[Bern, Dixon, Dunbar, Kosower] [Britto, Cachazo, Feng]

- In **on-shell configurations** of  $\ell_l$ , the integrand factorizes and produces a **cut equation** (partial fractioning the integrand!)

$$\sum_{\text{states } i \in T_\Gamma} \prod \mathcal{A}_i^{\text{tree}}(\ell_l^\Gamma) = \sum_{\substack{\Gamma' \geq \Gamma \\ k \in \bar{Q}_{\Gamma'}}} \frac{c_{\Gamma',k} m_{\Gamma',k}(\ell_l^\Gamma)}{\prod_{j \in (P_{\Gamma'} / P_\Gamma)} \rho_j(\ell_l^\Gamma)}$$

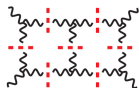


# Unitarity Approach to Computing Integrand Coefficients

[Bern, Dixon, Dunbar, Kosower] [Britto, Cachazo, Feng]

- In **on-shell configurations** of  $\ell_l$ , the integrand factorizes and produces a **cut equation** (partial fractioning the integrand!)

$$\sum_{\text{states } i \in T_\Gamma} \prod \mathcal{A}_i^{\text{tree}}(\ell_l^\Gamma) = \sum_{\substack{\Gamma' \supseteq \Gamma \\ k \in \bar{Q}_{\Gamma'}}} \frac{c_{\Gamma',k} m_{\Gamma',k}(\ell_l^\Gamma)}{\prod_{j \in (P_{\Gamma'} / P_\Gamma)} \rho_j(\ell_l^\Gamma)}$$



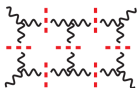
- Need **efficient computation** of (products of) **tree-level amplitudes**
  - **Off-shell recursion** relations [Berends, Giele 1988]
  - $D_s$ -dimensional **state sum**  $\rightarrow$  **Dimensional reduction** for resolving  $D_s$  dependence [Giele, Kunszt, Melnikov 2008]

# Unitarity Approach to Computing Integrand Coefficients

[Bern, Dixon, Dunbar, Kosower] [Britto, Cachazo, Feng]

- In **on-shell configurations** of  $\ell_l$ , the integrand factorizes and produces a **cut equation** (partial fractioning the integrand!)

$$\sum_{\text{states } i \in T_\Gamma} \prod \mathcal{A}_i^{\text{tree}}(\ell_l^\Gamma) = \sum_{\substack{\Gamma' \geq \Gamma \\ k \in \overline{Q}_{\Gamma'}}} \frac{c_{\Gamma',k} m_{\Gamma',k}(\ell_l^\Gamma)}{\prod_{j \in (P_{\Gamma'} / P_\Gamma)} \rho_j(\ell_l^\Gamma)}$$



- Need **efficient computation** of (products of) **tree-level amplitudes**
  - **Off-shell recursion** relations [Berends, Giele 1988]
  - $D_s$ -dimensional **state sum**  $\rightarrow$  **Dimensional reduction** for resolving  $D_s$  dependence [Giele, Kunszt, Melnikov 2008]

**Never construct** analytic integrand, numerics for every kinematic configuration

# Computing Products of Trees

- Analytic computations using **generalized unitarity** benefit from compact representations of tree-level amplitudes: **double copy**, **color-kinematics duality**, **rooted trees**, etc

# Computing Products of Trees

- Analytic computations using **generalized unitarity** benefit from compact representations of tree-level amplitudes: **double copy**, **color-kinematics duality**, **rooted trees**, etc
- For automated frameworks, dealing with numerics, **recursive** approaches (e.g. **BCFW**) are preferred as a way to make the tools more flexible

# Computing Products of Trees

- Analytic computations using **generalized unitarity** benefit from compact representations of tree-level amplitudes: **double copy**, **color-kinematics duality**, **rooted trees**, etc
- For automated frameworks, dealing with numerics, **recursive** approaches (e.g. **BCFW**) are preferred as a way to make the tools more flexible
- **Off-shell recursion** relations (Berends-Giele, Schwinger-Dyson) are appealing in particular because of their efficiency and generality

$$\text{Diagram} = \sum_{j=1}^{m-1} \text{Diagram}_j + \sum_{j=1}^{m-2} \sum_{k=j+1}^{m-1} \text{Diagram}_{j,k}$$

# Computing Products of Trees

- Analytic computations using **generalized unitarity** benefit from compact representations of tree-level amplitudes: **double copy**, **color-kinematics duality**, **rooted trees**, etc
- For automated frameworks, dealing with numerics, **recursive** approaches (e.g. **BCFW**) are preferred as a way to make the tools more flexible
- **Off-shell recursion** relations (Berends-Giele, Schwinger-Dyson) are appealing in particular because of their efficiency and generality

$$\text{Diagram} = \sum_{j=1}^{m-1} \text{Diagram}_1 + \sum_{j=1}^{m-2} \sum_{k=j+1}^{m-1} \text{Diagram}_2$$

We extended these relations to generic current types and to **products of trees**. Initial gravity applications benefited from a representation with **purely cubic interactions** of **Cheung and Remmen** [[arXiv:1705.00626](https://arxiv.org/abs/1705.00626)]

Compute the Integrand!

$$\sum_{\substack{\Gamma' \geq \Gamma \\ k \in \bar{Q}_{\Gamma'}}} \frac{c_{\Gamma',k} m_{\Gamma',k}(\ell_l^\Gamma)}{\prod_{j \in (P_{\Gamma'} / P_\Gamma)} \rho_j(\ell_l^\Gamma)} = \sum_{\text{states}} \prod_{i \in T_\Gamma} \mathcal{A}_i^{\text{tree}}(\ell_l^\Gamma)$$

# Solving Cut Equations

[Abreu, FFC, Ita, Jaquier, Page [arXiv:1703.05255]]



(a)



(b)



(c)



(d)



(e)



(f)



(g)

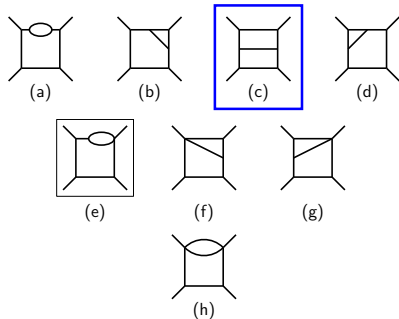


(h)

# Solving Cut Equations

For a maximal:

$$N\left(\begin{array}{c} \diagup \quad \diagdown \\ \text{---} \\ \diagdown \quad \diagup \end{array}, \ell_i^c\right) = R\left(\begin{array}{c} \diagup \quad \diagdown \\ \text{---} \\ \diagdown \quad \diagup \end{array}, \ell_i^c\right)$$



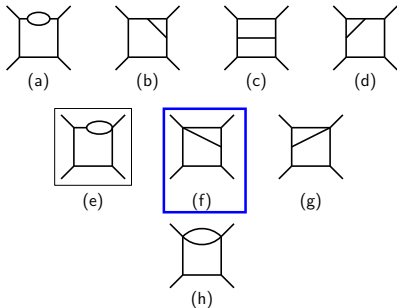
# Solving Cut Equations

For a maximal:

$$N\left(\text{Diagram (a)}, \ell_l^c\right) = R\left(\text{Diagram (a)}, \ell_l^c\right)$$

For a next-to-maximal:

$$N\left(\text{Diagram (f)}, \ell_l^f\right) = R\left(\text{Diagram (f)}, \ell_l^f\right) - \frac{1}{\rho_{fb}} N\left(\text{Diagram (e)}, \ell_l^f\right) - \frac{1}{\rho_{fc}} N\left(\text{Diagram (g)}, \ell_l^f\right)$$



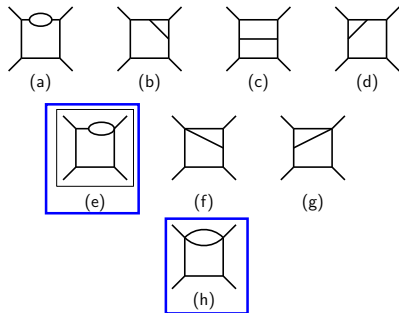
# Solving Cut Equations

For a maximal:

$$N\left(\text{Diagram (a)}, \ell_l^c\right) = R\left(\text{Diagram (a)}, \ell_l^c\right)$$

For a next-to-maximal:

$$N\left(\text{Diagram (e)}, \ell_l^f\right) = R\left(\text{Diagram (e)}, \ell_l^f\right) - \frac{1}{\rho_{fb}} N\left(\text{Diagram (f)}, \ell_l^f\right) - \frac{1}{\rho_{fc}} N\left(\text{Diagram (g)}, \ell_l^f\right)$$



And for the combined single-pole diagram an bubble-box:

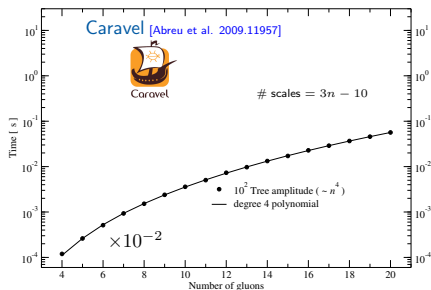
$$N\left(\text{Diagram (e)}, \ell_l^h\right) + \frac{1}{\rho_{he}} N\left(\text{Diagram (a)}, \ell_l^h\right) = R\left(\text{Diagram (e)}, \ell_l^h\right) - \frac{1}{\rho_{hf}} N\left(\text{Diagram (f)}, \ell_l^h\right) - \frac{1}{\rho_{hg}} N\left(\text{Diagram (g)}, \ell_l^h\right) - \frac{1}{(\rho_{he})^2} N\left(\text{Diagram (a)}, \ell_l^h\right) - \frac{1}{\rho_{hf}\rho_{fb}} N\left(\text{Diagram (e)}, \ell_l^h\right) - \frac{1}{\rho_{hf}\rho_{fc}} N\left(\text{Diagram (a)}, \ell_l^h\right) - \frac{1}{\rho_{hg}\rho_{gd}} N\left(\text{Diagram (f)}, \ell_l^h\right)$$

# Numerical Unitarity: Computational Complexity

Take as ex. **the numerical computation** of  $n$ -gluon color-ordered amps

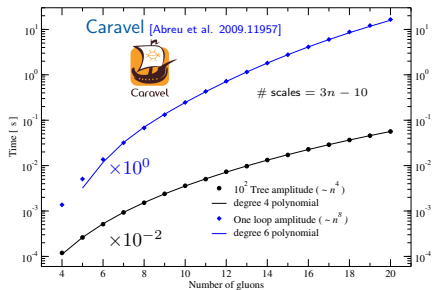
# Numerical Unitarity: Computational Complexity

Take as ex. **the numerical computation** of  $n$ -gluon color-ordered amps



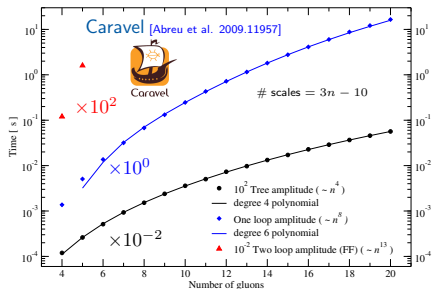
# Numerical Unitarity: Computational Complexity

Take as ex. the numerical computation of  $n$ -gluon color-ordered amps



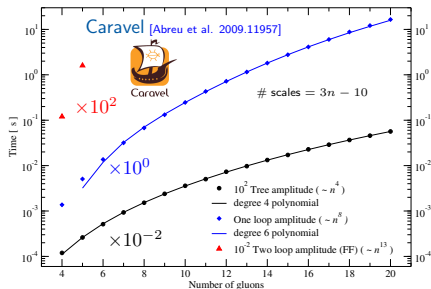
# Numerical Unitarity: Computational Complexity

Take as ex. the numerical computation of  $n$ -gluon color-ordered amps



# Numerical Unitarity: Computational Complexity

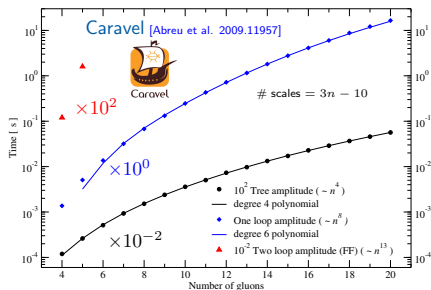
Take as ex. **the numerical computation** of  $n$ -gluon color-ordered amps



- **Polynomial complexity** to compute color-ordered amplitudes
- Dramatic computational increase in **loop order**
- **Asymptotic behavior** characterizes algorithm, but minimal impact in pheno
- **Combinatorial growth** in amplitudes needed for summed MEs

# Numerical Unitarity: Computational Complexity

Take as ex. **the numerical computation** of  $n$ -gluon color-ordered amps



- **Polynomial complexity** to compute color-ordered amplitudes
- Dramatic computational increase in **loop order**
- **Asymptotic behavior** characterizes algorithm, but minimal impact in pheno
- **Combinatorial growth** in amplitudes needed for summed MEs

**Analytic computations** for processes with not many scales (say 5 - 8) can considerably improve efficiency  $\rightarrow$  one can tame/handle the typical analytical **complexity growth** in **physical expressions!**

Amplitude **analytic expressions** found for:

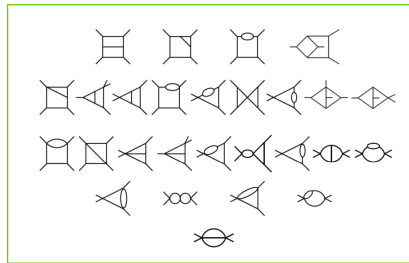
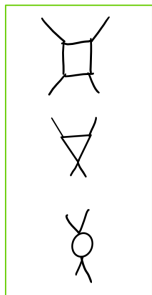
- selected 6-particle 1-loop amps
- essentially all 4-particle 2-loop amps
- all massless-QCD and selected massive 5-particle 2-loop amps
- selected 4-particle 3-loop amps

Amplitudes, IBP Identities, Numerical Unitarity

Exact Kinematics, Analytics from Numerics, Applications

# Numerical Stability

e.g. 4-gluon amplitudes



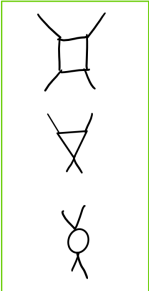
Function spaces with  
 $\mathcal{O}(10/50)$  dimensions

Function spaces with  
 $\mathcal{O}(100/1000)$  dimensions

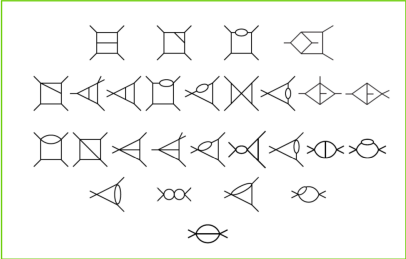
[Abreu, FFC, Ita, Jaquier, Page, Zeng, '17]

# Numerical Stability

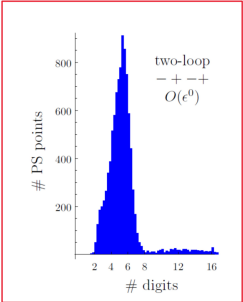
e.g. 4-gluon amplitudes



Function spaces with  $O(10/50)$  dimensions



Function spaces with  $O(100/1000)$  dimensions



- \* Relative precision of two-loop 4-gluon amp numerical calculation
- \* High-precision floating point arithmetic a remedy

[Abreu, FFC, Ita, Jaquier, Page, Zeng, '17]

## Numerical Unitarity in Generic Number Fields

\* (Almost) all steps to extract coefficients can be carried out with RATIONAL kinematics ( $\vec{x} \in \mathbb{Q}^n$ )

## Numerical Unitarity in Generic Number Fields

- \* (Almost) all steps to extract coefficients can be carried out with RATIONAL kinematics ( $\vec{x} \in \mathbb{Q}^n$ )
- \* But RATIONAL computer algebra reflects the corresponding ANALYTIC COMPLEXITY!

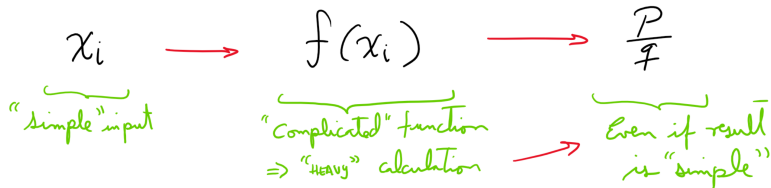
## Numerical Unitarity in Generic Number Fields

- \* (Almost) all steps to extract coefficients can be carried out with RATIONAL kinematics ( $\vec{x} \in \mathbb{Q}^n$ )
- \* But RATIONAL computer algebra reflects the corresponding ANALYTIC COMPLEXITY!

$$x_i \longrightarrow f(x_i) \longrightarrow \frac{P}{Q}$$

# Numerical Unitarity in Generic Number Fields

- \* (Almost) all steps to extract **coefficients** can be carried out with **RATIONAL** kinematics ( $\vec{x} \in \mathbb{Q}^n$ )
- \* But **RATIONAL** computer algebra reflects the corresponding **ANALYTIC COMPLEXITY!**



# Finite (Number) Fields

\* Take integers  $\mathbb{F}_p = \{0, 1, 2, \dots, p-1\}$  where  $p$  is prime

# Finite (Number) Fields

\* Take integers  $\mathbb{F}_p = \{0, 1, 2, \dots, p-1\}$  where  $p$  is prime

\* Perform binary operations modulo  $p$

$$5 + 7 \pmod{11} = 1 ; 5 \times 7 \pmod{11} = 2 ; 5 - 7 \pmod{11} = 9$$

# Finite (Number) Fields

- \* Take integers  $\mathbb{F}_p = \{0, 1, 2, \dots, p-1\}$  where  $p$  is prime
- \* Perform binary operations modulo  $p$   
 $5 + 7 \pmod{11} = 1$  ;  $5 \times 7 \pmod{11} = 2$  ;  $5 - 7 \pmod{11} = 9$
- \* Every  $a \in \mathbb{F}_p$  has multiplicative inverse, so  $\mathbb{F}_p$  is a field  
 $5^{-1} \pmod{11} = 9$  (unique!)

# Finite (Number) Fields

- \* Take integers  $\mathbb{F}_p = \{0, 1, 2, \dots, p-1\}$  where  $p$  is prime
- \* Perform binary operations modulo  $p$   
 $5+7 \pmod{11} = 1$  ;  $5 \times 7 \pmod{11} = 2$  ;  $5-7 \pmod{11} = 9$
- \* Every  $a \in \mathbb{F}_p$  has multiplicative inverse, so  $\mathbb{F}_p$  is a field  
 $5^{-1} \pmod{11} = 9$  (unique!)
- \* All rational operations possible (but no square roots, etc)

# Finite (Number) Fields

- \* Take integers  $\mathbb{F}_p = \{0, 1, 2, \dots, p-1\}$  where  $p$  is prime
- \* Perform binary operations modulo  $p$   
 $5+7 \bmod 11 = 1$  ;  $5 \times 7 \bmod 11 = 2$  ;  $5-7 \bmod 11 = 9$
- \* Every  $a \in \mathbb{F}_p$  has multiplicative inverse, so  $\mathbb{F}_p$  is a field  
 $5^{-1} \bmod 11 = 9$  (unique!)
- \* All rational operations possible (but no square roots, etc)
- \* From  $\mathbb{Q}$  to  $\mathbb{F}_p$  :  
 $a = \frac{r}{s} \in \mathbb{Q} \Rightarrow a \bmod p = r \cdot (s^{-1} \bmod p) \bmod p$

\* MAP  $\mathbb{Q}^m$  into  $\mathbb{F}_p^m$  and try to reconstruct result!

- \* MAP  $\mathbb{Q}^m$  into  $\mathbb{F}_p^m$  and try to reconstruct result!
- \* If cardinality  $p$  is smaller than CPU's word size ( $2^{64}$ ) operations will be very fast

# From $\mathbb{Q}^n$ to $\mathbb{F}^n$ (& Back!)

[von Manteuffel, Schabinger [arXiv:1406.4513]]

\* MAP  $\mathbb{Q}^m$  into  $\mathbb{F}_p^m$  and try to reconstruct result!

\* If *cardinality*  $p$  is smaller than CPU's word size ( $2^{64}$ )  
operations will be very fast

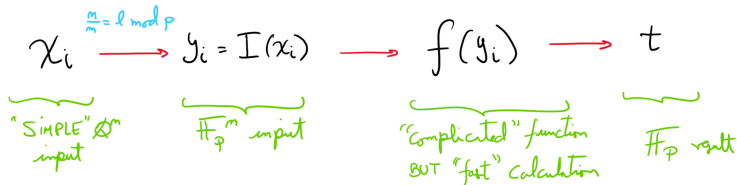
$$x_i \xrightarrow{\frac{m}{m} = l \bmod p} y_i = I(x_i) \longrightarrow f(y_i) \longrightarrow t$$

# From $\mathbb{Q}^n$ to $\mathbb{F}^n$ (& Back!)

[von Manteuffel, Schabinger [arXiv:1406.4513]]

\* **MAP**  $\mathbb{Q}^m$  into  $\mathbb{F}_p^m$  and try to reconstruct result!

\* If **cardinality**  $p$  is smaller than CPU's word size ( $2^{64}$ ) operations will be very fast

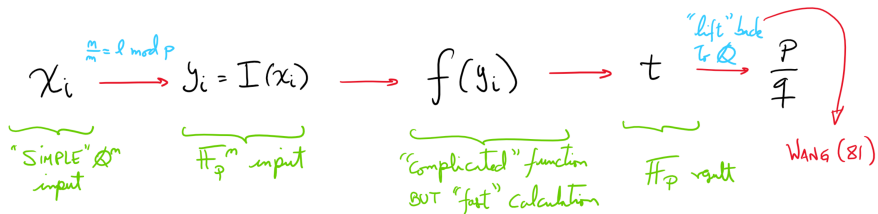


# From $\mathbb{Q}^n$ to $\mathbb{F}^n$ (& Back!)

[von Manteuffel, Schabinger [arXiv:1406.4513]]

\* MAP  $\mathbb{Q}^m$  into  $\mathbb{F}_p^m$  and try to reconstruct result!

\* If *cardinality*  $p$  is smaller than CPU's word size ( $2^{64}$ ) operations will be very fast



## From Exact Evaluations to Analytics

$$A(l_e) = \sum_{\Gamma, i} c_{\Gamma, i} \frac{m_{\Gamma, i}(l_e)}{\prod_{k \in \Gamma} \rho_k(l_e)}$$

## From Exact Evaluations to Analytics

$$A(l_e) = \sum_{\Gamma, i} c_{\Gamma, i} \frac{m_{\Gamma, i}(l_e)}{\prod_{k \in \Gamma} \beta_k(l_e)} \rightarrow c_{\Gamma, i} = c_{\Gamma, i}(\beta)$$

# From Exact Evaluations to Analytics

$$A(l_e) = \sum_{\Gamma, i} C_{\Gamma, i} \frac{m_{\Gamma, i}(l_e)}{\prod_{k \in \Gamma} \rho_k(l_e)} \rightarrow C_{\Gamma, i} = C_{\Gamma, i}(\beta)$$

Such that  $C_{\Gamma, i}$  is a rational function of  $\beta$

$$C_{\Gamma, i} = \frac{\sum_j f_j(x_k) \beta^{j+N}}{\sum_j g_j \beta^{j+M}}$$

} STRUCTURE NOT KNOWN & PRIORI !

$\beta$  can represent an  $x_e$ , dimensional parameter  $\varepsilon$ , etc

# From Exact Evaluations to Analytics

$$A(l_e) = \sum_{\Gamma, i} C_{\Gamma, i} \frac{m_{\Gamma, i}(l_e)}{\prod_{k \in \Gamma} p_k(l_e)} \rightarrow C_{\Gamma, i} = C_{\Gamma, i}(\beta)$$

Such that  $C_{\Gamma, i}$  is a rational function of  $\beta$

$$C_{\Gamma, i} = \frac{\sum_j f_j(x_k) \beta^{j+N}}{\sum_j q_j \beta^{j+M}}$$

} STRUCTURE NOT KNOWN & PRIORI !

$\beta$  can represent an  $x_e$ , dimensional parameter  $\varepsilon$ , etc

Sampling  $C_{\Gamma, i}$  over multiple  $\beta$  values allows to determine the unknown  $f_j(x_k)$ ,  $q_j$

# Fitting Ansatzes from Numeric Samples

## THIELE'S INTERPOLATION FORMULA:

Every rational function can be written as a *continued fraction*

$$f(x) = \frac{\sum_{r=0}^R n_r x^r}{\sum_{r'=0}^{R'} d_{r'} x^{r'}} = a_0 + \frac{x - y_0}{a_1 + \frac{x - y_1}{a_2 + \frac{x - y_2}{\dots + \frac{x - y_{N-1}}{a_N}}}}$$

- \* Determine  $a_i$  by *evaluating*  $f(y_i)$  ( $y_i$  random)
- \* Stop when  $f(y_{i+1})$  *matches* interpolated value (+ error checks)
- \* Through only *field operations* recover rational function  
(FF's result can be lifted to  $\mathbb{Q}$ )

# Fitting Ansatzes from Numeric Samples

## THIELE'S INTERPOLATION FORMULA:

Every rational function can be written as a *continued fraction*

$$f(x) = \frac{\sum_{r=0}^R n_r x^r}{\sum_{r'=0}^{R'} d_{r'} x^{r'}} = a_0 + \frac{x - y_0}{a_1 + \frac{x - y_1}{a_2 + \frac{x - y_2}{\dots + \frac{x - y_{N-1}}{a_N}}}}$$

- \* Determine  $a_i$  by *evaluating*  $f(y_i)$  ( $y_i$  random)
- \* Stop when  $f(y_{i+1})$  *matches* interpolated value (+ error checks)
- \* Through only *field operations* recover rational function  
(FF's result can be lifted to  $\mathbb{Q}$ )

This technology has been **extended in many directions**, allowing e.g. **multivariate rational reconstruction**, fitting of physics-aware **ansatze**, performing numerical expansions using **p-adic numbers**, exploiting **partial fractions** in reconstruction procedure, and much more

# The CARAVEL Framework

A framework to explore multi-loop multi-leg scattering amplitudes in the SM and beyond



# The CARAVEL Framework

A framework to explore multi-loop multi-leg scattering amplitudes in the SM and beyond

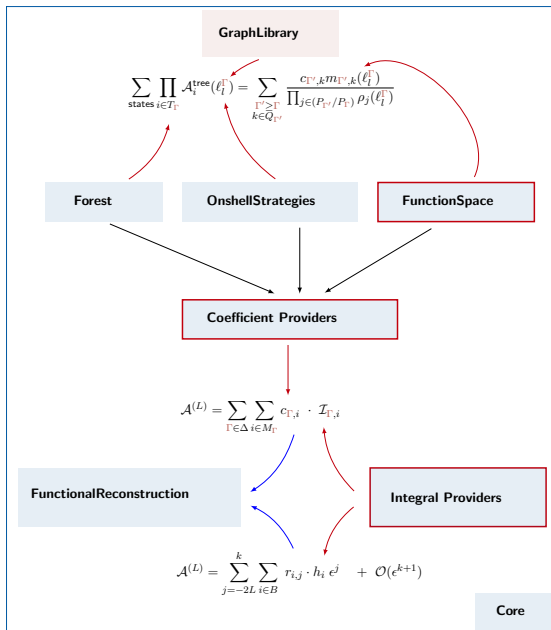
- A modular C++17 library implementing the multi-loop numerical unitarity method

[Abreu, Dormans, FFC, Ita, Kraus, Page, Pascual, Ruf, Sotnikov, arXiv:2009.11957]

- Numerics in (high-precision) floating-point, rational and modular arithmetic
- Generic design for calculations in QFT, e.g. in the SM, gravity theories, and more
- Algebraic tools for semi-analytical calculations in C++
- Publicly available @ GitLab [Link!]



# The CARAVEL Framework



Includes general tools for:

- $D$ -dimensional kinematics
- graph isomorphism techniques
- tree-level and multi-loop cut calculations
- Generic scattering-plane integrand parametrizations
- Selected master-surface decompositions
- on-shell phase-space parametrizations
- Feynman integral handling
- Algebraic tools
- ...

Caravel @ GitLab:

<https://gitlab.com/caravel-public/caravel>



# Outlook

- pQFT gives us a **systematically improvable** framework to make predictions for hadron collider and gravitational-wave observatory: **lots** of scattering amplitudes needed!
- Exploiting **analytic properties** we can compute scattering amplitudes efficiently. As an example presented the **multi-loop numerical unitarity** method
- Exact numerical evaluations have become a main tool to reconstruct corresponding **analytic expressions** of very complex amplitudes
- We look forward to **increase our understanding** of the fundamental interactions among elementary particles

# Outlook

- pQFT gives us a **systematically improvable** framework to make predictions for hadron collider and gravitational-wave observatory: **lots** of scattering amplitudes needed!
- Exploiting **analytic properties** we can compute scattering amplitudes efficiently. As an example presented the **multi-loop numerical unitarity** method
- Exact numerical evaluations have become a main tool to reconstruct corresponding **analytic expressions** of very complex amplitudes
- We look forward to **increase our understanding** of the fundamental interactions among elementary particles

Thanks!