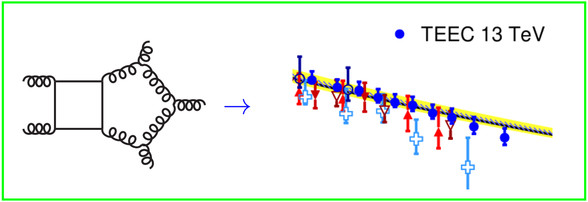
Numerical Techniques for Gravity Scattering Amplitudes

The Numerical Unitarity Method

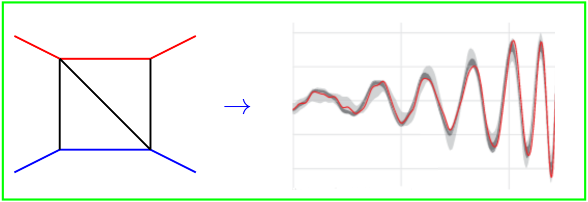


Fernando Febres Cordero
Department of Physics, Florida State University

Gravitational Waves Meet Amplitudes, ICTP-SAIFR, Sao Paulo
8/24/2023

Numerical Techniques for Gravity Scattering Amplitudes

The Numerical Unitarity Method



Fernando Febres Cordero
Department of Physics, Florida State University

Gravitational Waves Meet Amplitudes, ICTP-SAIFR, Sao Paulo
8/24/2023

Outline

(Very Quick) Introduction

The Numerical Unitarity Method for Multi-Loop Amplitudes

Beyond Numerics: Exact Kinematics

Truncation and Underlying Amplitudes

$$\sigma_{h_1 h_2 \to H} = \alpha_s^{\kappa} \left(\sigma_{\rm LO} + \alpha_s \; \sigma_{\rm NLO} + \alpha_s^2 \; \sigma_{\rm NNLO} + \alpha_s^3 \; \sigma_{\rm N^3LO} + \cdots \right)$$

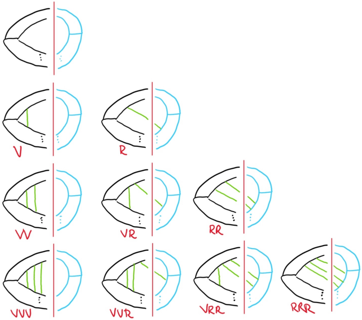
typical th uncertainties

LO: $\sim$ α_s^{κ} — $\sim 50\%$

NLO: $+$ $\alpha_s^{\kappa+1}$ — $\sim 17\%$

NNLO: $+$ $\alpha_s^{\kappa+2}$ — $\sim 7\%$

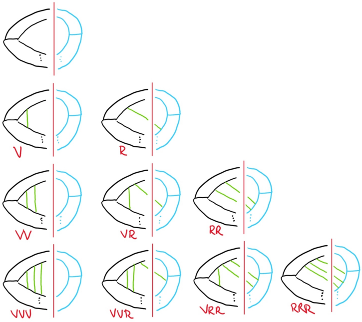
N^{3}LO: $+$ $\alpha_s^{\kappa+3}$ — few $\%$



Truncation and Underlying Amplitudes

$$\sigma_{h_1 h_2 \to H} = \alpha_s^\kappa \left(\sigma_{\mathrm{LO}} + \alpha_s\, \sigma_{\mathrm{NLO}} + \alpha_s^2\, \sigma_{\mathrm{NNLO}} + \alpha_s^3\, \sigma_{\mathrm{N^3LO}} + \cdots \right)$$

typical th uncertainties

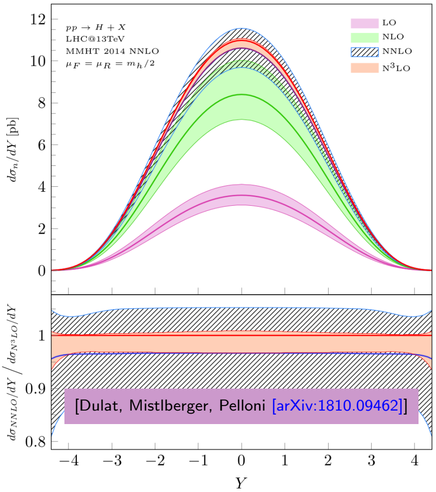


LO:	$\sim$ α_s^κ	$\sim 50\%$
NLO:	$+$ $\alpha_s^{\kappa+1}$	$\sim 17\%$
NNLO:	$+$ $\alpha_s^{\kappa+2}$	$\sim 7\%$
$\mathrm{N^3LO}$:	$+$ $\alpha_s^{\kappa+3}$	few $\%$

A myriad of amplitudes are required for precision calculations

pQCD in Action

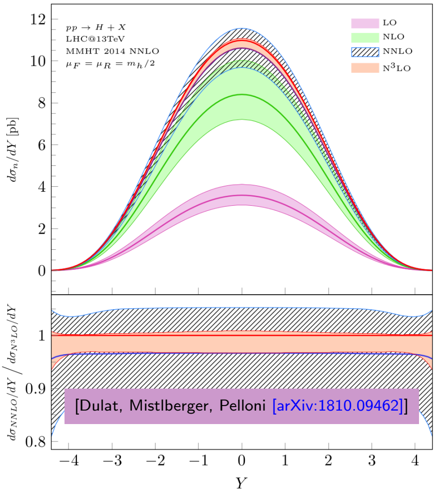
- Higgs boson production at the LHC
- Differential in rapidity
- Convergence achieved apparent at fourth order



[Dulat, Mistlberger, Pelloni [arXiv:1810.09462]]

pQCD in Action

- Higgs boson production at the LHC
- Differential in rapidity
- Convergence achieved apparent at fourth order



[Dulat, Mistlberger, Pelloni [arXiv:1810.09462]]

NNLO QCD a basic requirement for a variety of multi-particle/multi-jet processes in years to come! Stresses our computation capabilities

Outline

(Very Quick) Introduction

The Numerical Unitarity Method for Multi-Loop Amplitudes

Beyond Numerics: Exact Kinematics

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

The numerical unitarity method avoids issues by:

- Performing reduction and evaluation simultaneously
- Working numerically

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 [Smirnov, Petukhov; Bitoun, Bogner, Klausen, Panzer]

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 [Smirnov, Petukhov; Bitoun, Bogner, Klausen, Panzer]

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 $\mathcal{Q}_\Gamma = \{m_{\Gamma,k}(\ell_l) | k \in Q_\Gamma\}$ 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 [Smirnov, Petukhov; Bitoun, Bogner, Klausen, Panzer]

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

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

Integrand Parametrizations

- Given diagram Γ make an *adaptive* momentum parametrization

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

Integrand Parametrizations

- Given diagram Γ make an *adaptive* momentum parametrization

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

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

Integrand Parametrizations

- Given diagram Γ make an *adaptive* momentum parametrization

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

- **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, Page, Zeng [arXiv:1703.05273]; see also Ossola, Papadopoulos, Pittau; Bobadilla, Mastrolia, Peraro, Primo]

Integrand Parametrizations

- Given diagram Γ make an *adaptive* momentum parametrization

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

- **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, Page, Zeng [arXiv:1703.05273]; see also Ossola, Papadopoulos, Pittau; Bobadilla, Mastrolia, Peraro, Primo]

We have automated these integrand parametrizations. Typically less than $\mathcal{O}(5\%)$ (IBP-)reducible monomials (with scattering-plane variables) remain!

Integrand Parametrizations

- Given diagram Γ make an *adaptive* momentum parametrization

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

- **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, Page, Zeng [arXiv:1703.05273]; see also Ossola, Papadopoulos, Pittau; Bobadilla, Mastrolia, Peraro, Primo]
- **Master-Surface Basis**: powerful parametrization trivializing map to master integrals [Ita, arXiv:1510.05626]. That is, $Q_\Gamma = M_\Gamma \cup S_\Gamma$:

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

Master/Surface Decompositions [Ita '15; Abreu, **FFC**, Ita, Page, Zeng '17]

Consider the integration by parts (IBP) relation on Γ

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

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

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

Master/Surface Decompositions

[Ita '15; Abreu, **FFC**, Ita, Page, Zeng '17]

Consider the integration by parts (IBP) relation on Γ

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

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

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

Write ansatz for u_j^ν expanded in external and loop momenta, and solve polynomial equations using algebraic geometry techniques

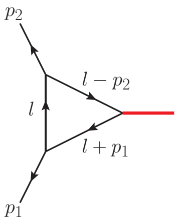
Using the u_j^ν we can build a full set of surface terms and fill the rest of the space with master integrands

Related [Boehm, Georgoudis, Larsen, Schulze, Zhang '16 - '19]
[Agarwal, von Manteuffel '19]

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):

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

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

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

A 1-loop Example for Surface Terms: Part 2

Now we have the surface term:

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

The scalar triangle integrand can be replaced by a surface term, though commonly it is kept in 1-loop calculations, keeping it as a "master" integral.

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

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

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)

Surface Terms Factory

[Abreu, **FFC**, Ita, Jaquier, Page, Ruf, Sotnikov [arXiv:2002.12374]]

Solutions to u_j^ν are universal. 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 $t_r(\ell_l)$ (e.g. the scattering-plane tensors), and using the vector $t_r(\ell_l)u_j^\nu$:

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

Surface Terms Factory

[Abreu, **FFC**, Ita, Jaquier, Page, Ruf, Sotnikov [arXiv:2002.12374]]

Solutions to u_j^ν are universal. 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 $t_r(\ell_l)$ (e.g. the scattering-plane tensors), and using the vector $t_r(\ell_l) u_j^\nu$:

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

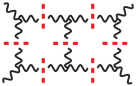
A four-graviton amplitude calculation in Einstein-Hilbert gravity structurally the same as a four-gluon amplitude calculation in QCD!

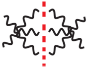
Unitarity Approach to Computing Integrand Coefficients

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

- In on-shell configurations of ℓ_l, the integrand factorizes and produces a *cut equation*:

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



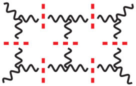


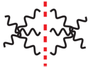
Unitarity Approach to Computing Integrand Coefficients

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

- In on-shell configurations of ℓ_l, the integrand factorizes and produces a *cut equation*:

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





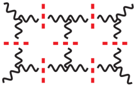
- Need efficient computation of (products of) tree-level amplitudes
  - Off-shell recursion relations [Berends, Giele 1988]
  - D_s-dimensional state sum $\rightarrow$ Dimensional reduction for resolving D_s dependence [Giele, Kunszt, Melnikov 2008]

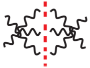
Unitarity Approach to Computing Integrand Coefficients

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

- In on-shell configurations of ℓ_l, the integrand factorizes and produces a *cut equation*:

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



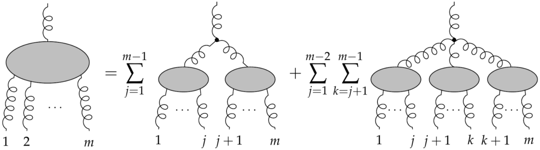


- Need efficient computation of (products of) tree-level amplitudes
  - Off-shell recursion relations [Berends, Giele 1988]
  - D_s-dimensional state sum $\to$ Dimensional reduction for resolving D_s dependence [Giele, Kunszt, Melnikov 2008]

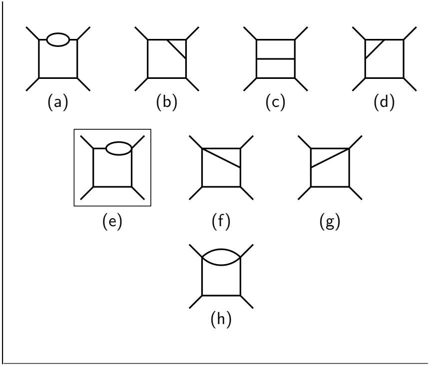
Never construct analytic integrand, numerics for every kinematic configuration

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
- For automated frameworks, dealing with numerics, recursive approaches are preferred as a way to make the tools more flexible
- Off-shell recursion relations (Berends-Giele, Schwinger-Dyson) are appealing in particular because their efficiency



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]

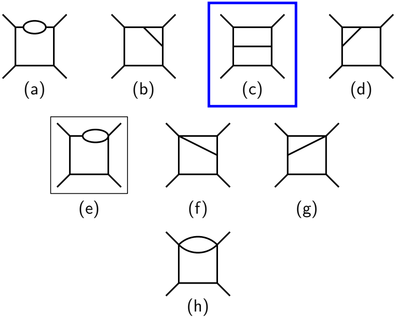

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Solving Cut Equations

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

For a maximal:

$$N\left(\boxminus, \ell_l^{\mathrm{c}}\right) \quad = \quad R\left(\boxminus, \ell_l^{\mathrm{c}}\right)$$



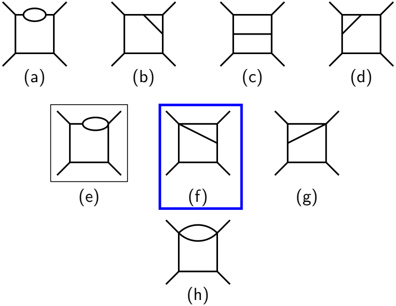
For a maximal:

$N\Big($  $, \ell_l^{c}\Big) \quad = \quad R\Big($  $, \ell_l^{c}\Big)$

For a next-to-maximal:

$N\Big($  $, \ell_l^{f}\Big) = R\Big($  $, \ell_l^{f}\Big)$

$-\frac{1}{\rho_{\mathrm{fb}}} N\Big($  $, \ell_1^{f}\Big) - \frac{1}{\rho_{\mathrm{fc}}} N\Big($  $, \ell_l^{f}\Big)$



Solving Cut Equations

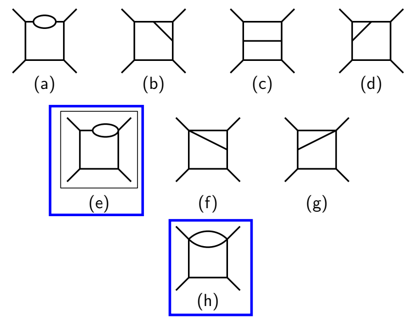
[Abreu, **FFC**, Ita, Jaquier, Page [arXiv:1703.05255]]

For a maximal:

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

For a next-to-maximal:

$$N\left(\text{(f)}, \ell_l^{\rm f}\right) = R\left(\text{(f)}, \ell_l^{\rm f}\right) - \frac{1}{\rho_{\rm fb}} N\left(\text{(b)}, \ell_1^{\rm f}\right) - \frac{1}{\rho_{\rm fc}} N\left(\text{(c)}, \ell_l^{\rm f}\right)$$



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

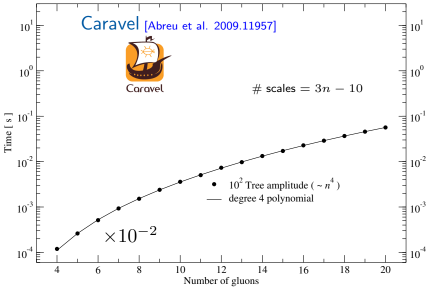
$$N\left(\text{(h)}, \ell_l^{\rm h}\right) + \frac{1}{\rho_{\rm he}} N\left(\text{(e)}, \ell_l^{\rm h}\right) = R\left(\text{(h)}, \ell_l^{\rm h}\right) - \frac{1}{\rho_{\rm hf}} N\left(\text{(f)}, \ell_l^{\rm h}\right) - \frac{1}{\rho_{\rm hg}} N\left(\text{(g)}, \ell_l^{\rm h}\right) - \frac{1}{(\rho_{\rm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

Numerical Unitarity: Computational Complexity

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

Numerical Unitarity: Computational Complexity

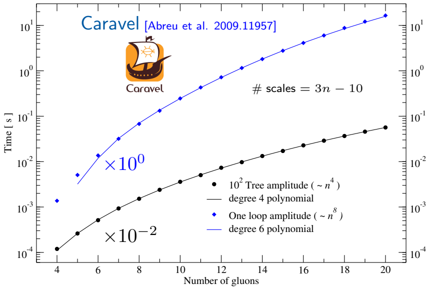
Take as ex. the numerical computation of n-gluon color-ordered amps



- 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

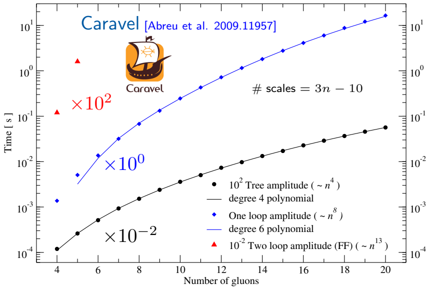
Take as ex. the numerical computation of n-gluon color-ordered amps



- 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

Numerical Unitarity: Computational Complexity

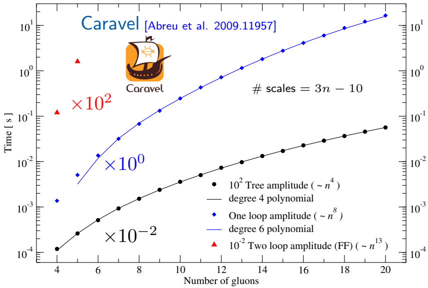
Take as ex. the numerical computation of n-gluon color-ordered amps



- 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 $\longrightarrow$ tame/handle the typical analytical exponential complexity growth!

Amplitude analytic expressions found for:

- selected 6-particle 1-loop amps
- essentially all 4-particle 2-loop amps
- selected 5-particle 2-loop amps
- selected 4-particle 3-loop amps

Outline

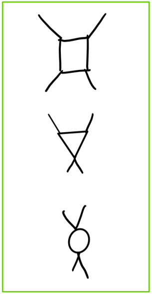
(Very Quick) Introduction

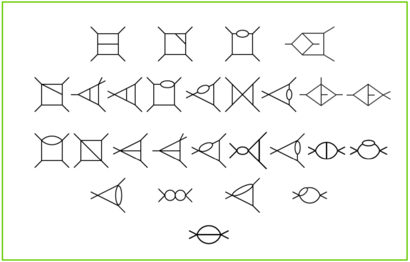
The Numerical Unitarity Method for Multi-Loop Amplitudes

Beyond Numerics: Exact Kinematics

Numerical Stability

e.g. 4-gluon amplitudes





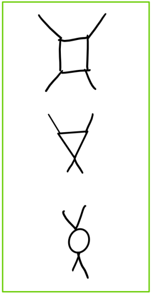
Function spaces with $\mathcal{O}(10/50)$ dimensions

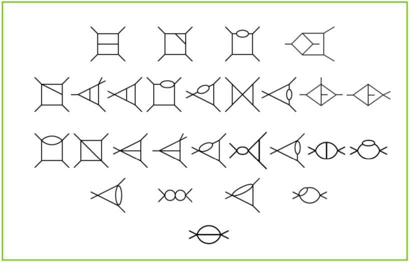
Function spaces with $\mathcal{O}(100/1000)$ dimensions

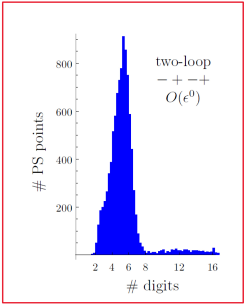
[Abreu, FFC, Ita, Jaquier, Page, Zeng, '17]

Numerical Stability

e.g. 4-gluon amplitudes







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

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

$$x_i \longrightarrow f(x_i) \longrightarrow \frac{p}{q}$$

"simple" input

"complicated" function
⇒ "HEAVY" calculation →

Even if result is "simple"

Finite (Number) Fields

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

Finite (Number) Fields

* Take integers $\mathbb{F}_p = \{1, 2, \ldots, p-1\}$ where p is prime
* Perform binary operations modulo p

  $5 + 7 \mod 11 = 1$; $5 \times 7 \mod 11 = 2$; $5 - 7 \mod 11 = 9$

Finite (Number) Fields

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

  $5+7 \mod 11 = 1$; $5 \times 7 \mod 11 = 2$; $5 - 7 \mod 11 = 9$
* Every $a \in \mathbb{F}_p$ has multiplicative inverse, so $\mathbb{F}_p$ is a field

  $5^{-1} \mod 11 = 9$ (unique!)

Finite (Number) Fields

* Take integers $\mathbb{F}_p = \{1, 2, \ldots, p-1\}$ where p is prime
* Perform binary operations modulo p

  $5 + 7 \mod 11 = 1$; $5 \times 7 \mod 11 = 2$; $5 - 7 \mod 11 = 9$

* Every $a \in \mathbb{F}_p$ has multiplicative inverse, so $\mathbb{F}_p$ is a field

  $5^{-1} \mod 11 = 9$ (unique!)

* All rational operations possible (but no square roots, etc)

Finite (Number) Fields

* Take integers $\mathbb{F}_p = \{1, 2, \ldots, p-1\}$ where p is prime
* Perform binary operations modulo p

$5+7 \mod 11 = 1$; $5 \times 7 \mod 11 = 2$; $5-7 \mod 11 = 9$

* Every $a \in \mathbb{F}_p$ has multiplicative inverse, so $\mathbb{F}_p$ is a field

$5^{-1} \mod 11 = 9$ (unique!)

* All rational operations possible (but no square roots, etc)
* From $\mathbb{Q}$ to $\mathbb{F}_p$:

$a = \frac{r}{s} \in \mathbb{Q} \Rightarrow a \mod p = r \cdot (s^{-1} \mod p) \mod p$

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

[von Manteuffel, Schabinger [arXiv:1406.4513]]

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

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

[von Manteuffel, Schabinger [arXiv:1406.4513]]

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

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

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

[von Manteuffel, Schabinger [arXiv:1406.4513]]

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

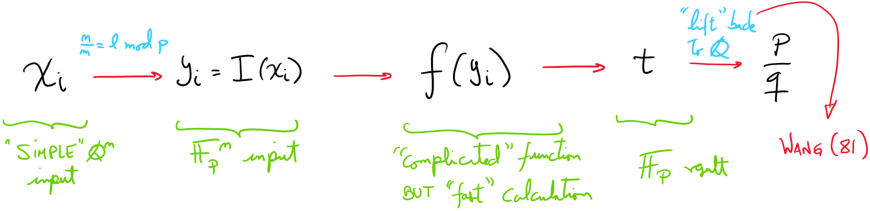
$$x_i \xrightarrow{\frac{n}{m} = \ell \bmod p} y_i = I(x_i) \longrightarrow f(y_i) \longrightarrow t$$

- x_i: "SIMPLE" $\mathbb{Q}^n$ input
- $y_i = I(x_i)$: $\mathbb{F}_p^n$ input
- $f(y_i)$: "Complicated" function BUT "fast" calculation
- t: $\mathbb{F}_p$ result

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

[von Manteuffel, Schabinger [arXiv:1406.4513]]

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



From Exact Evaluations to Analytics

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

From Exact Evaluations to Analytics

$$\mathcal{A}(\ell_e) = \sum_{\Gamma,i} c_{\Gamma,i} \frac{m_{\Gamma,i}(\ell_e)}{\prod_{k\in\Gamma} \rho_k(\ell_e)} \longrightarrow \quad c_{\Gamma,i} = c_{\Gamma,i}(\beta)$$

From Exact Evaluations to Analytics

$$\mathcal{A}(\ell_e) = \sum_{\Gamma,i} c_{\Gamma,i} \frac{m_{\Gamma,i}(\ell_e)}{\prod_{k\in\Gamma} \rho_k(\ell_e)} \longrightarrow c_{\Gamma,i} = c_{\Gamma,i}(\beta)$$

Such that $c_{\Gamma,i}$ is a rational function of β

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

STRUCTURE NOT KNOWN A PRIORI !

β can represent an x_ℓ, dimensional parameter ε, etc

From Exact Evaluations to Analytics

$$\mathcal{A}(\ell_e) = \sum_{\Gamma,i} c_{\Gamma,i} \frac{m_{\Gamma,i}(\ell_e)}{\prod_{k\in\Gamma} \rho_k(\ell_e)} \longrightarrow c_{\Gamma,i} = c_{\Gamma,i}(\beta)$$

Such that $c_{\Gamma,i}$ is a rational function of β

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

STRUCTURE NOT KNOWN A PRIORI !

β can represent an x_ℓ, dimensional parameter ε, etc

Sampling $c_{\Gamma,i}$ over multiple β values allows to determine the unknow $f_j(x_k),\ q_j$

Fitting Ansatze from Numeric Samples

THIELE'S INTERPOLATION FORMULA:

Every rational function can be written as a continued fraction

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

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

Fitting Ansatze from Numeric Samples

THIELE'S INTERPOLATION FORMULA:

Every rational function can be written as a continued fraction

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

* Determine a_i by evaluating $f(y_i)$ (y_i random)
* Stop when $f(y_{i+1})$ matches interpolated value (+ extra 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

Caravel

The Caravel Framework

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

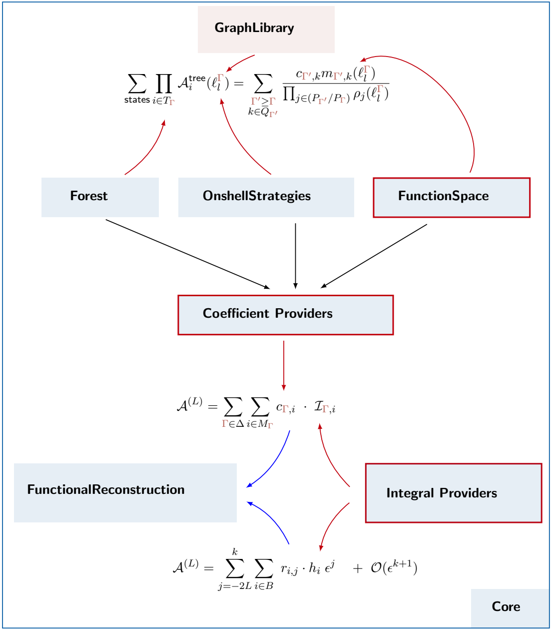
- A modular C++17 library implementing the multi-loop numerical unitarity method

  [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!

Caravel

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

Applications to Gravity Scattering Amplitudes

Several applications to the analytic computation of 5-particle scattering amplitudes in the Standard Model

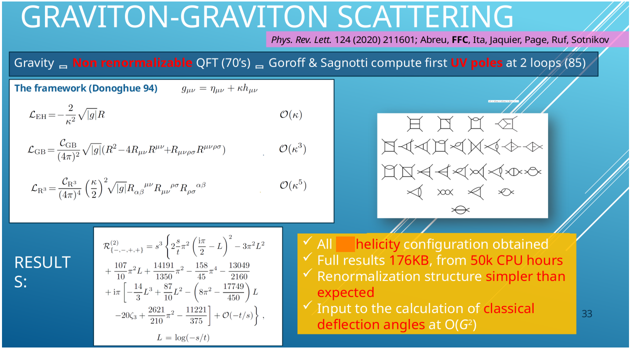
Applications to Gravity Scattering Amplitudes

Several applications to the analytic computation of 5-particle scattering amplitudes in the Standard Model

- Analytic form of the two-loop graviton-graviton scattering amplitudes

Several applications to the analytic computation of 5-particle scattering amplitudes in the Standard Model

- Analytic form of the two-loop graviton-graviton scattering amplitudes



Applications to Gravity Scattering Amplitudes

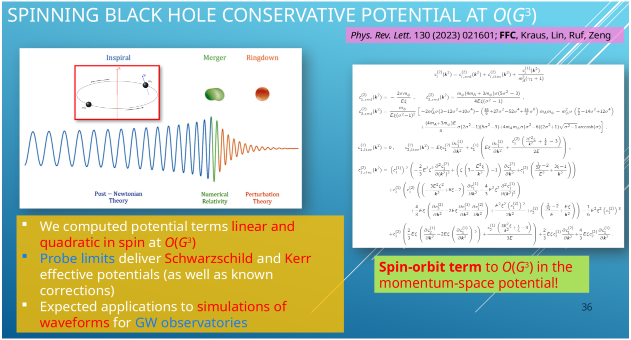
Several applications to the analytic computation of 5-particle scattering amplitudes in the Standard Model

- Analytic form of the two-loop graviton-graviton scattering amplitudes
- Computed to third post-Minkowskian order conservative binary dynamics including up-to S^2 terms $\rightarrow$ See Manfred Kraus talk

Applications to Gravity Scattering Amplitudes

Several applications to the analytic computation of 5-particle scattering amplitudes in the Standard Model

- Analytic form of the two-loop graviton-graviton scattering amplitudes
- Computed to third post-Minkowskian order conservative binary dynamics including up-to S^2 terms → See Manfred Kraus talk



Outlook

Outlook

- Presented the numerical unitarity method for the computation of multi-loop scattering amplitudes
- Usage of exact kinematics and computations allow extraction of analytic expressions for the amplitudes (*A new standard?!*)
- We have released the Caravel framework which contains a large set of tools to carry these calculations
- Numerical untarity is well suited for calculations in gravity and we look forward to further applications, higher spins, finite-size effects, more loops, etc

Outlook

- Presented the numerical unitarity method for the computation of multi-loop scattering amplitudes
- Usage of exact kinematics and computations allow extraction of analytic expressions for the amplitudes (*A new standard?!*)
- We have released the Caravel framework which contains a large set of tools to carry these calculations
- Numerical untarity is well suited for calculations in gravity and we look forward to further applications, higher spins, finite-size effects, more loops, etc

Thanks!