Automatic calculation of one-loop amplitudes

Andreas van Hameren

IFJ-PAN, Kraków

in collaboration with

C.G. Papadopoulos, NCSR “Demokritos”, Athens

R. Pittau, University of Granada

RADCOR 2009, Ascona, 27-10-2009

Supported in part by the EU RTN European Programme, MRTN-CT-2006-035505 (HEPTOOLS, Tools and Precision Calculations for Physics Discoveries at Colliders) and by the Polish Ministry of Scientific Research and Information Technology grant No 153/6 PR UE/2007/7 2007-2010.

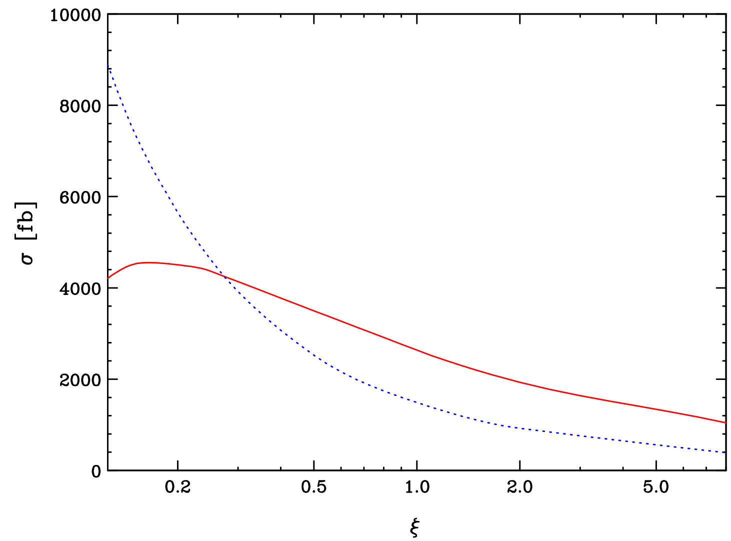
Motivation

- physics at LHC demands precise qualitative knowledge about signals and backgrounds;
- Monte Carlo programs are the preferred tools to condensate such knowledge;
- multi-leg hard processes need to be included in these. Many interesting signals (Higgs production) and their backgrounds include multi-particle final states.
- NLO corrections have to be included
  - to reduce scale dependence;
  - to get better description of shapes of distributions;

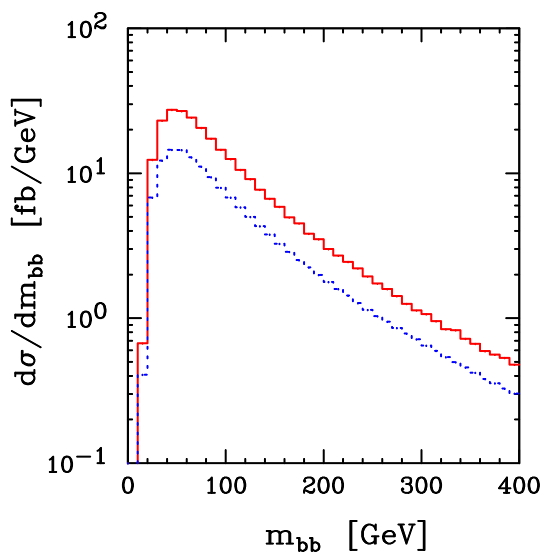
Motivation

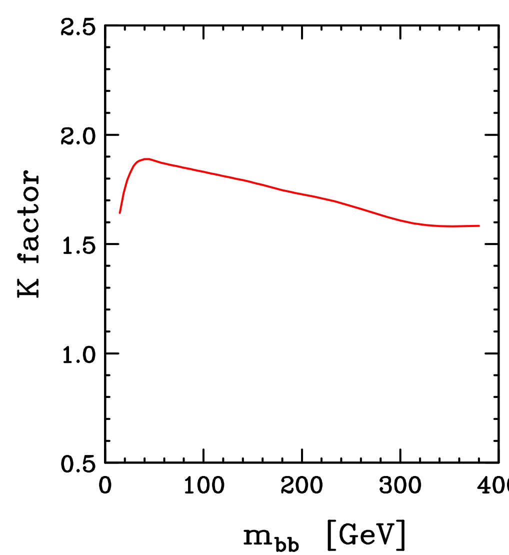
LO vs. NLO $pp \to t\bar{t}b\bar{b}$ (Bevilacqua,Czakon,Papadopoulos,Pittau,Worek)

scale dependence:



shape:





Motivation

Backgrounds

- $pp \to t\bar{t}Z$ Lazopoulos,Melnikov,Petriello
- $pp \to t\bar{t} + j$ Dittmaier,Uwer,Weinzierl
- $pp \to VV + j$ Dittmaier,Kallweit,Uwer; Campbell,Ellis,Zanderighi
- $pp \to t\bar{t}b\bar{b}$ Bredenstein,Denner,Dittmaier,Pozzorini; Bevilacqua,Czakon,Papadopoulos,Pittau,Worek
- $pp \to VVV$ ZZZ:Lazopoulos,Melnikov,Petriello; WWZ:Hankele,Zeppenfeld; VVV: Binoth,Ossola,Papadopoulos,Pittau
- $pp \to VV + 2j$ VBF: Jäger,Oleari,Zeppenfeld; Bozzi
- $pp \to W + 3j$ Ellis,Melnikov,Zanderighi; Berger,Bern,Dixon,Febres Cordero,Forde,Gleisberg,Ita,Kosower,Maître

Signals

- $pp \to H + 2j$ Campbell,Ellis,Zanderighi; Ciccolini,Denner,Dittmaier
- $pp \to H + t\bar{t}$ Beenakker,Dittmaier,Krämer,Plümer,Spira,Zerwas; Dawson,Jackson,Reina,Wackeroth

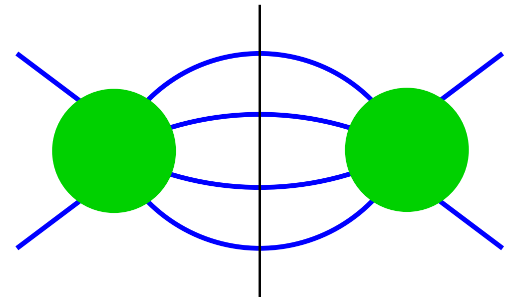
Motivation

- LO calculations (including partonic phase space generation) have been completely automated;
- One of the parts to be automated for NLO is the calculation of the one-loop amplitude necessary for the virtual contribution.
  - FeynArts/FormCalc Hahn
  - Golem Binoth,Guffanti,Guillet,Heinrich,Karg,Kauer,Reiter,Reuter
  - Grace Belanger,Boudjema,Fujimoto,Ishikawa,Kaneko,Kato,Shimizu
  - Rocket Giele,Zanderighi
  - BlackHat Berger,Bern,Dixon,Febres Cordero,Forde,Ita,Kosower,Maître
  - D-dim Unitarity Lazopoulos
- we wanted to build a tool that can be readily integrated in the existing fully automatic LO-system Helac/Phegas.

Ingredients for NLO calculations

LO calculation

$$\langle O \rangle^{\text{LO}} = \int d\Phi_n \, |\mathcal{M}_n^{(0)}|^2 \, O_n^{\text{LO}} \quad =$$

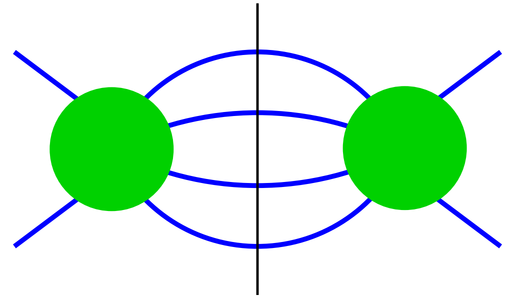


- Observable O_n^{LO} represents some interesting distribution, and includes phase space cuts;
- $\mathcal{M}_n^{(0)}$ is the Born (tree-level) matrix element.

Ingredients for NLO calculations

LO calculation

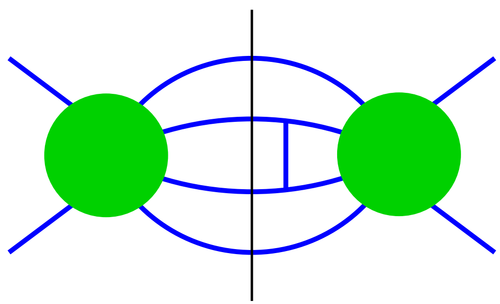
$$\langle O\rangle^{\mathrm{LO}} = \int d\Phi_n\, |\mathcal{M}_n^{(0)}|^2\, O_n^{\mathrm{LO}} \quad =$$



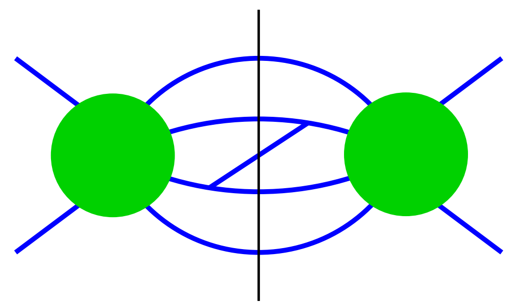
NLO calculation

$$\langle O\rangle^{\mathrm{NLO}} = \int d\Phi_n\, 2\Re\left(\mathcal{M}_n^{(0)}\mathcal{M}_n^{(1)}\right) O_n^{\mathrm{LO}} + \int d\Phi_{n+1}\, |\mathcal{M}_{n+1}^{(0)}|^2\, O_{n+1}^{\mathrm{NLO}}$$

$$=$$



$$+$$



- $\mathcal{M}_n^{(1)}$ is the one-loop amplitude
- $\mathcal{M}_{n+1}^{(0)}$ is the real-radiation (tree-level) matrix element;
- O_{n+1}^{NLO} includes a jet algorithm;

Ingredients for NLO calculations

Phase space integration

has to be done by Monte Carlo.

Helicity summation

is best performed explicitly by a numerical sum over squared helicity amplitudes, maybe even by Monte Carlo.

Color summation

is best performed explicitly by a numerical sum over squared colorful amplitudes, eventually necessarily by Mont Carlo.

$$\langle O\rangle = \int d\Phi_n(\{p\}_n) \sum_{\{\lambda\}_n} \sum_{\{a\}_n} |\mathcal{M}_n(\{p\}_n, \{\lambda\}_n, \{a\}_n)|^2\, O_n(\{p\}_n)$$

Ingredients for NLO calculations

Phase space integration

has to be done by Monte Carlo.

Helicity summation

is best performed explicitly by a numerical sum over squared helicity amplitudes, maybe even by Monte Carlo.

Color summation

is best performed explicitly by a numerical sum over squared colorful amplitudes, eventually necessarily by Mont Carlo.

Efficiently in the color flow representation

$$\mathcal{M}^{i_1,i_2,\dots,i_n}_{j_1,j_2,\dots,j_n} = (t^{a_1})^{i_1}_{j_1}(t^{a_2})^{i_2}_{j_2}\cdots(t^{a_n})^{i_n}_{j_n}\mathcal{M}(a_1,a_2,\dots,a_n)$$
$$= \sum_{\sigma\in\text{perm}} \delta^{i_{\sigma(1)}}_{j_{\sigma(2)}}\delta^{i_{\sigma(2)}}_{j_{\sigma(3)}}\cdots\delta^{i_{\sigma(n)}}_{j_{\sigma(1)}}\,\mathcal{A}(\sigma(1),\sigma(2),\dots,\sigma(n))$$

Calculation of $\mathcal{A}$ relatively easy, product of δ-s often equal zero.

Amplitude calculation

Traditional approach

- determine Feynman graphs;
- apply Feynman rules;
- get an expression, in terms of invariants, helicities,...etc
- simplify the expression as much as possible;
- write a numerical computer program to evaluate this expression.

Amplitude calculation

Traditional approach

- determine Feynman graphs;
- apply Feynman rules;
- get an expression, in terms of invariants, helicities,...etc
- simplify the expression as much as possible;
- write a numerical computer program to evaluate this expression.

Problem:

for multi-particle amplitudes the expressions become huge.

Amplitude calculation

Traditional approach

- determine Feynman graphs;
- apply Feynman rules;
- get an expression, in terms of invariants, helicities,...etc
- simplify the expression as much as possible;
- write a numerical computer program to evaluate this expression.

Problem:

for multi-particle amplitudes the expressions become huge.

Solution:

avoid expressions, and take a numerical approach from the start.

Numerical amplitude calculation

LSZ-formula: amplitude = connected Green function with external propagators replaced by spinors/polarization vectors.

Dyson-Schwinger equation: relates connected Green functions with different numbers of exteral legs.

ϕ^3-theory

$$-i(p^2 - m^2)G_{n+1}(p, p_1, \ldots, p_n) =$$

$$g \int dp_b \, \delta(p - p_a - p_b) \Bigg[\sum_{\{j\}} G_{k+1}(p_a, p_{j_1}, \ldots, p_{j_k}) \, G_{n-k+1}(p_b, p_{j_{k+1}}, \ldots, p_{j_n})$$

$$+ \frac{1}{2} G_{n+2}(p_a, p_b, p_1, \ldots, p_n) \Bigg]$$

Numerical amplitude calculation

LSZ-formula: amplitude = connected Green function with external propagators replaced by spinors/polarization vectors.

Dyson-Schwinger equation: relates connected Green functions with different numbers of exteral legs.

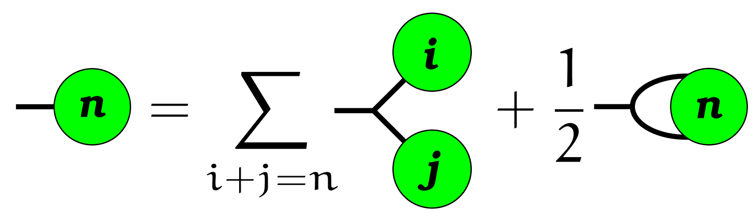
ϕ^3-theory

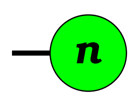
$$-i(p^2 - m^2)G_{n+1}(p, p_1, \ldots, p_n) =$$

$$g \int dp_b \, \delta(p - p_a - p_b) \left[\sum_{\{j\}} G_{k+1}(p_a, p_{j_1}, \ldots, p_{j_k}) \, G_{n-k+1}(p_b, p_{j_{k+1}}, \ldots, p_{j_n}) + \frac{1}{2} G_{n+2}(p_a, p_b, p_1, \ldots, p_n) \right]$$

Off−shell currents:

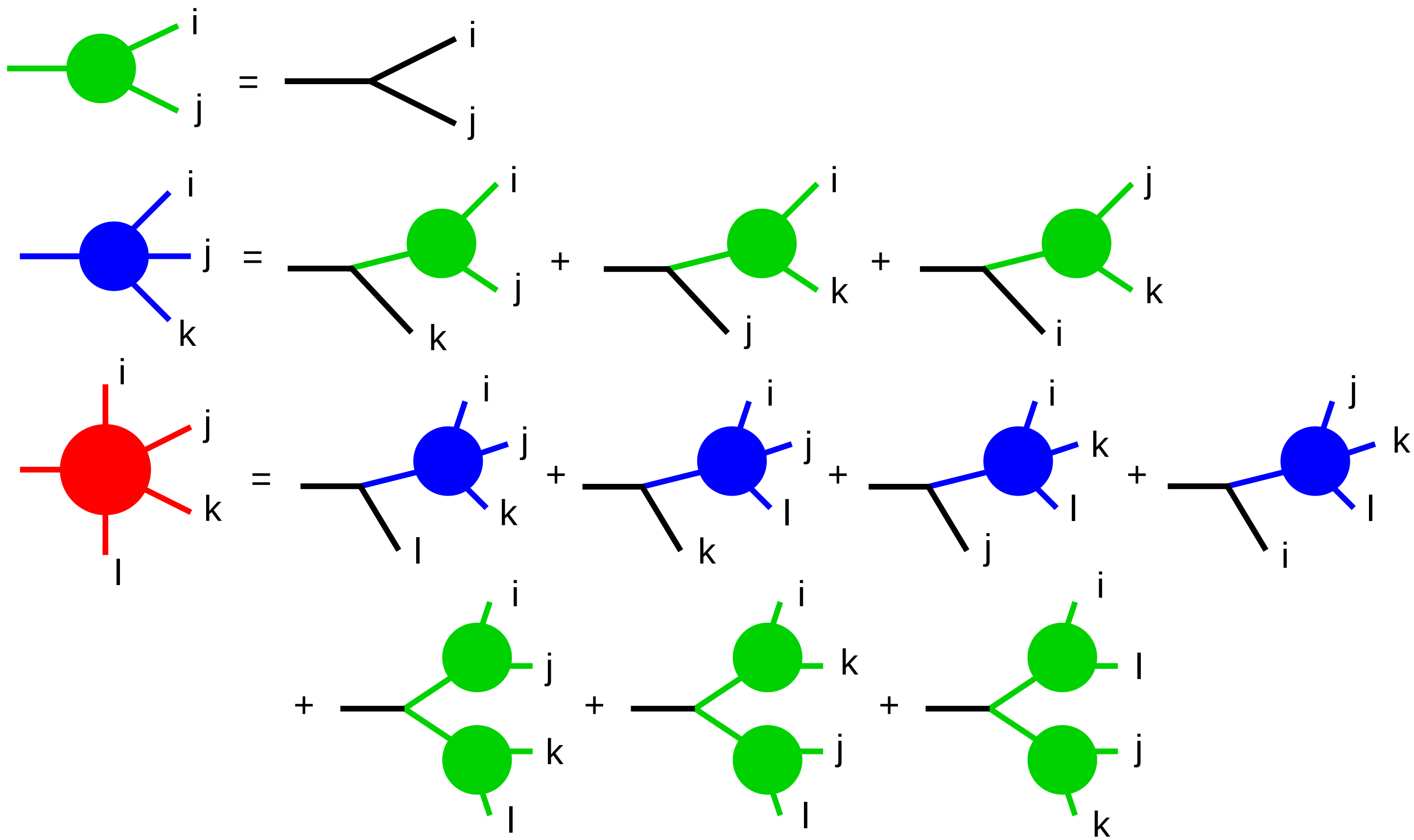
replace all propagators refering to external particles by spinors/polarization vectors



Analytic solution:  = sum of Feynman graphs.

Calculation of tree-level amplitudes

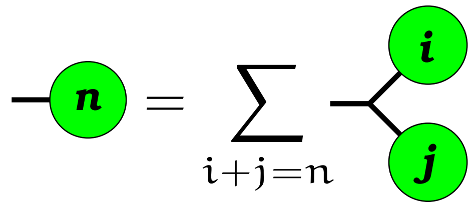
Numerical Dyson-Schwinger approach Berends,Giele'88; Caravaglios,Moretti'95:



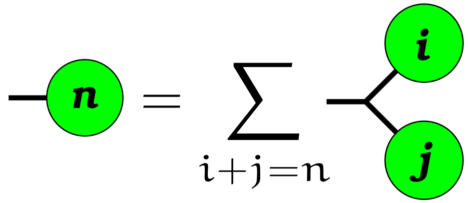
- each graph represents a set numbers for the field components;
- Efficient: $O(n!)$ for graphs to $O(3^n)$, $n =$ number of exteral legs;
- Straightforward to automatize.

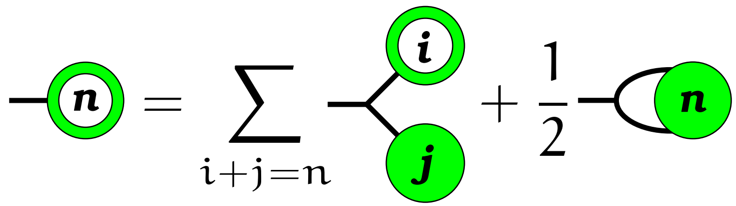
Calculation of one-loop amplitudes

Tree-level recursion:



Calculation of one-loop amplitudes

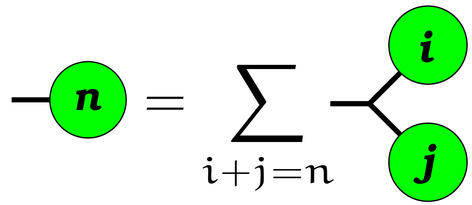
Tree-level recursion: 

One-loop recursion: 

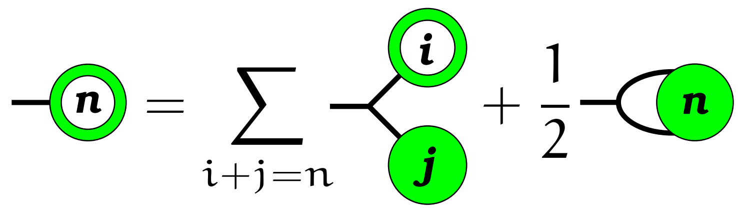
Solid blobs are tree−like, blobs with a hole contain one loop.

Calculation of one-loop amplitudes

Tree-level recursion:



One-loop recursion:

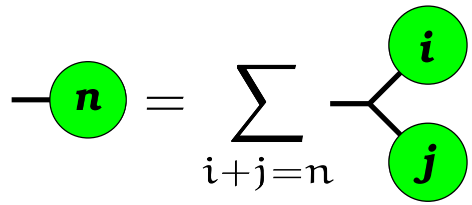


Solid blobs are tree−like, blobs with a hole contain one loop.

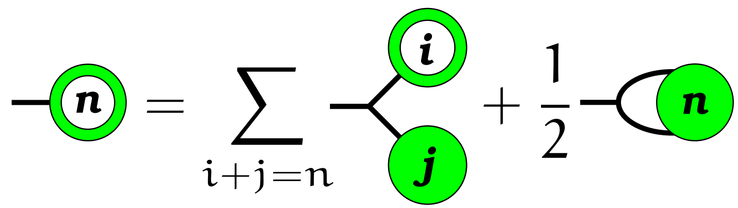
How to deal with the loop integration?

Calculation of one-loop amplitudes

Tree-level recursion:



One-loop recursion:

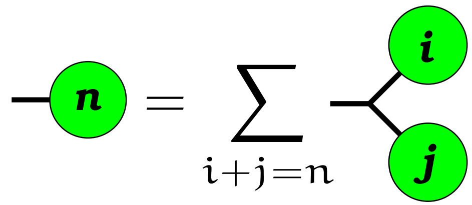


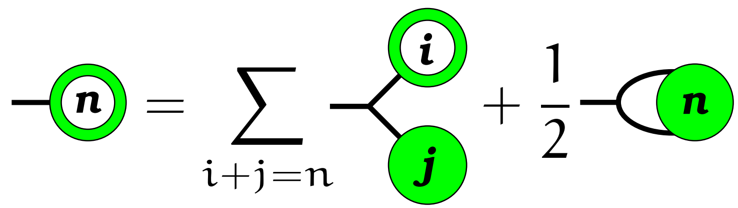
Solid blobs are tree−like, blobs with a hole contain one loop.

How to deal with the loop integration?

Expand the process dependent one-loop object in terms of universal one-loop objects, *eg* tensor integrals, scalar integrals.

Calculation of one-loop amplitudes

Tree-level recursion: 

One-loop recursion: 

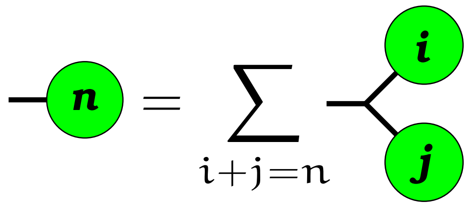
Solid blobs are tree−like, blobs with a hole contain one loop.

How to deal with the loop integration?
Expand the process dependent one-loop object in terms of universal one-loop objects, *eg* tensor integrals, scalar integrals.

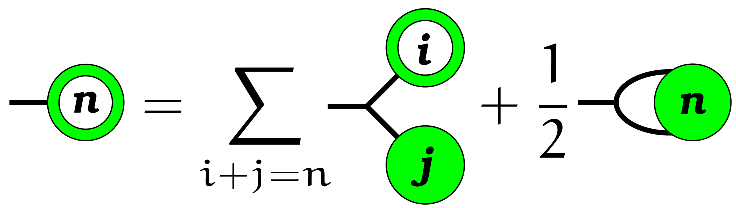
- derive recursive relations for the coefficients (AvH);

Calculation of one-loop amplitudes

Tree-level recursion:



One-loop recursion:



Solid blobs are tree−like, blobs with a hole contain one loop.

How to deal with the loop integration?
Expand the process dependent one-loop object in terms of universal one-loop objects, *eg* tensor integrals, scalar integrals.

- derive recursive relations for the coefficients (AvH);
- more practical is to forget about one-loop recursion, make optimal use of tree-level recursion to get coefficients for full amplitude.

One-loop amplitude with O_{ssola}P_{apadopoulos}P_{ittau}

Identify a set of n_{tot} denominators and write

$$\mathcal{M}^{(1)} = \sum_{I \subset \{0,1,2,\dots,n_{\text{tot}}-1\}} \int d^{\text{Dim}}q \, \frac{N_I(q)}{\prod_{i \in I} D_i} \quad , \quad D_i = (q + p_i)^2 - m_i^2$$

One-loop amplitude with OPP

Identify a set of n_{tot} denominators and write

$$\mathcal{M}^{(1)} = \sum_{I\subset\{0,1,2,\ldots,n_{\text{tot}}-1\}} \int d^{\text{Dim}}q\, \frac{N_I(q)}{\prod_{i\in I} D_i} \quad , \quad D_i = (q+p_i)^2 - m_i^2$$

Integrals can be expressed in terms of universal scalar-functions:

$$\int \frac{d^{\text{Dim}}q\; N(q)}{D_0 D_1 \cdots D_{n-1}} = \sum_{i_1,i_2,i_3,i_4} d_{i_1 i_2 i_3 i_4} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2} D_{i_3} D_{i_4}} + \sum_{i_1,i_2,i_3} c_{i_1 i_2 i_3} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2} D_{i_3}}$$

$$+ \sum_{i_1,i_2} b_{i_1 i_2} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2}} + \sum_{i_1} a_{i_1} \int \frac{d^{\text{Dim}}q}{D_{i_1}} + \text{rational terms} + \mathcal{O}(\text{Dim}-4)$$

One-loop amplitude with OPP

Identify a set of n_{tot} denominators and write

$$\mathcal{M}^{(1)} = \sum_{I\subset\{0,1,2,\dots,n_{\text{tot}}-1\}} \int d^{\text{Dim}}q \, \frac{N_I(q)}{\prod_{i\in I} D_i} \quad , \quad D_i = (q+p_i)^2 - m_i^2$$

Integrals can be expressed in terms of universal scalar-functions:

$$\int \frac{d^{\text{Dim}}q \; N(q)}{D_0 D_1 \cdots D_{n-1}} = \sum_{i_1,i_2,i_3,i_4} d_{i_1 i_2 i_3 i_4} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2} D_{i_3} D_{i_4}} + \sum_{i_1,i_2,i_3} c_{i_1 i_2 i_3} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2} D_{i_3}}$$

$$+ \sum_{i_1,i_2} b_{i_1 i_2} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2}} + \sum_{i_1} a_{i_1} \int \frac{d^{\text{Dim}}q}{D_{i_1}} + \text{rational terms} + \mathcal{O}(\text{Dim}-4)$$

Coefficients can be determined from polynomial equations involving few more coefficients

$$\frac{N(q)}{D_0 D_1 \cdots D_{n-1}} = \sum_{i_1,i_2,i_3,i_4} \frac{d_{i_1 i_2 i_3 i_4} + \tilde{d}_{i_1 i_2 i_3 i_4}(q)}{D_{i_1} D_{i_2} D_{i_3} D_{i_4}} + \sum_{i_1,i_2,i_3} \frac{c_{i_1 i_2 i_3} + \tilde{c}_{i_1 i_2 i_3}(q)}{D_{i_1} D_{i_2} D_{i_3}}$$

$$+ \sum_{i_1,i_2} \frac{b_{i_1 i_2} + \tilde{b}_{i_1 i_2}(q)}{D_{i_1} D_{i_2}} + \sum_{i_1} \frac{a_{i_1} + \tilde{a}_{i_1}(q)}{D_{i_1}}$$

1 extra coefficient for $\tilde{d}$, 6 for $\tilde{c}$, 8 for $\tilde{b}$, 4 for $\tilde{a}$

One-loop amplitude with OPP

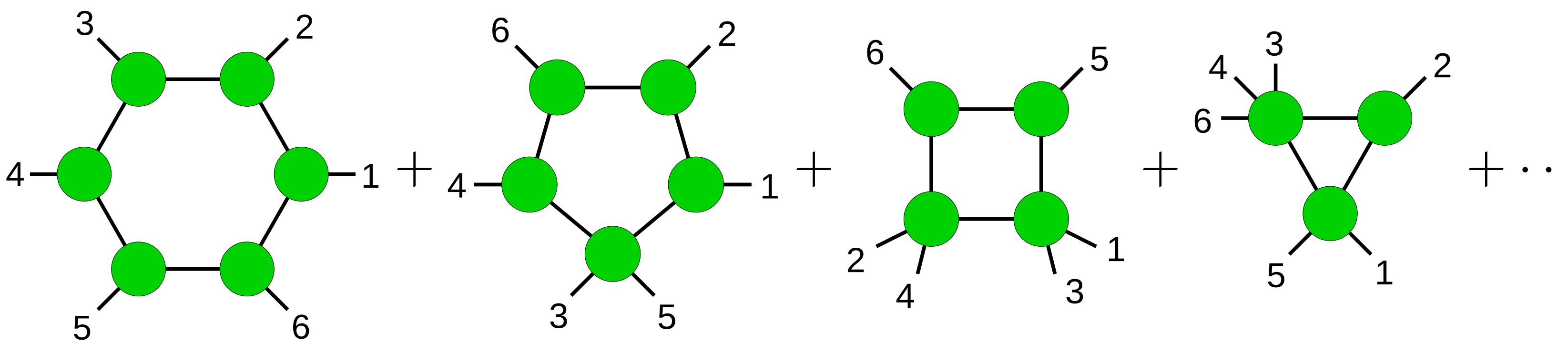
$$\int \frac{d^{\text{Dim}}q\, N(q)}{D_0 D_1 \cdots D_{n-1}} = \sum_{i_1,i_2,i_3,i_4} d_{i_1 i_2 i_3 i_4} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2} D_{i_3} D_{i_4}} + \sum_{i_1,i_2,i_3} c_{i_1 i_2 i_3} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2} D_{i_3}}$$
$$+ \sum_{i_1,i_2} b_{i_1 i_2} \int \frac{d^{\text{Dim}}q}{D_{i_1} D_{i_2}} + \sum_{i_1} a_{i_1} \int \frac{d^{\text{Dim}}q}{D_{i_1}} + \text{rational terms} + O(\text{Dim} - 4)$$

- universal set of scalar-functions can be coded once and for all eg. QCDloop/FF Ellis,Zanderighi,van Oldenborgh, OneLOop AvH;
- coefficients d, c, b, a can be determined numerically from polynomial equations in 4 dimensions.
- rational terms can be written in terms of
  - simple universal integrals with already determined coefficients (R_1, coming from denominators for $\text{Dim} \neq 4$),
  - plus a finite counterterm (R_2, coming from numerator for $\text{Dim} \neq 4$) Draggiotis,Garzelli,Malamos,Papadopoulos,Pittau.
- to NLO we are not interested in $O(\text{Dim} - 4)$.

Evaluation of the numerator

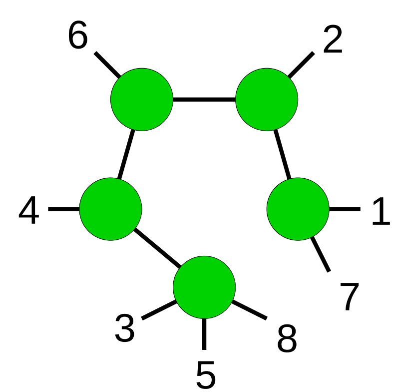
We choose to go explicitly through all possible 1PI structures:

$$\mathcal{M}^{(1)} = \sum_{I\subset\{0,1,2,\dots,n_{\text{tot}}-1\}} \int d^{\text{Dim}}q \, \frac{N_I(q)}{\prod_{i\in I} D_i}$$

$=$ 

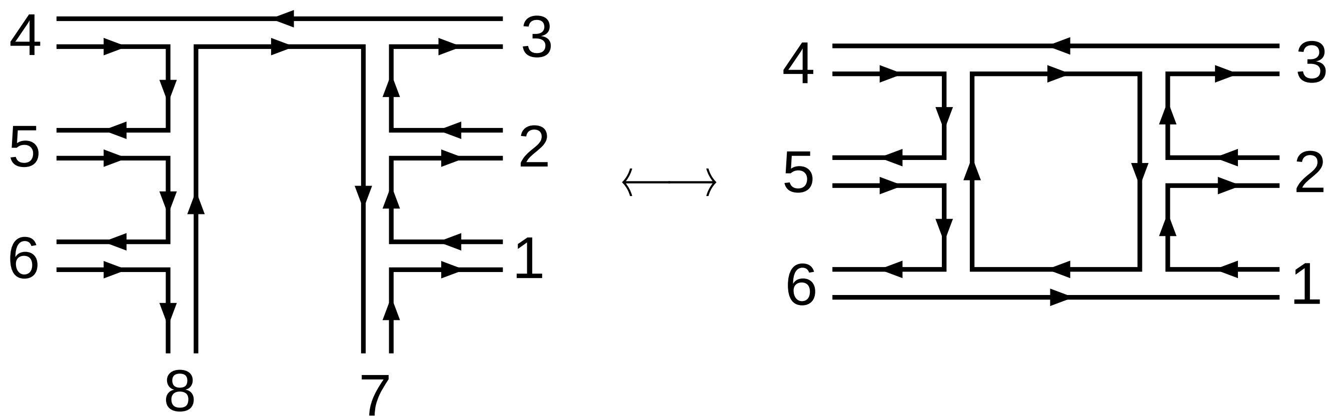
All blobs are tree-level, without denominators containing the integration momentum.

The numerator can be obtained for each term from a tree-level amplitude with 2 more particles, and restricted such that it contains the necessary propagators.



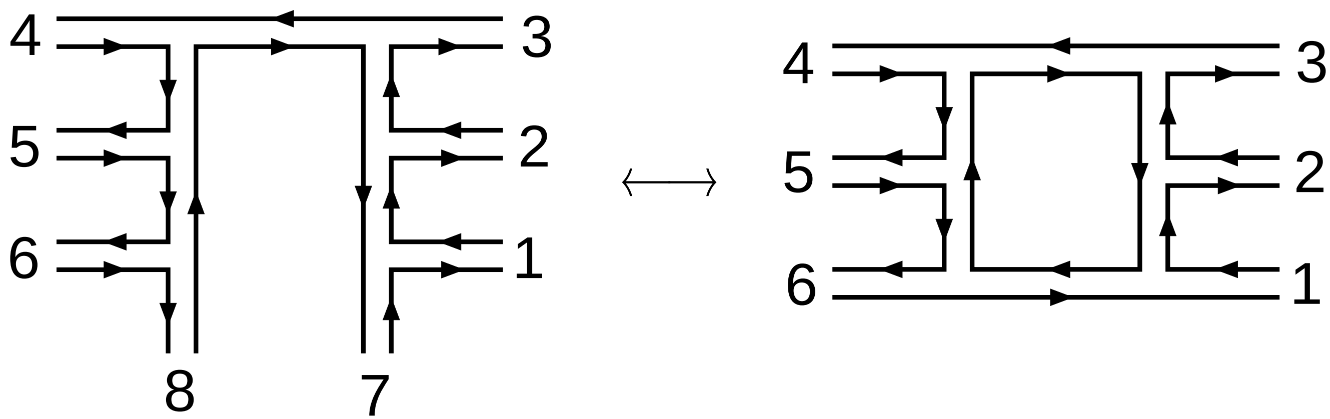
Evaluation of the numerator

One-loop amplitude in the color-flow representation

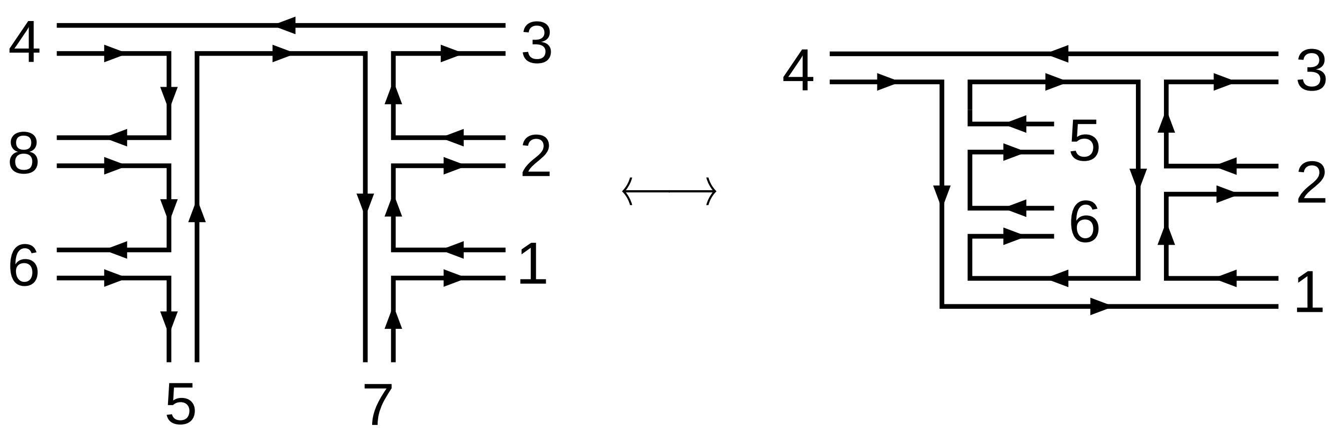


Evaluation of the numerator

One-loop amplitude in the color-flow representation



Including the extra particle in the sum over permutations gives the non-planar contributions



The program Helac-1Loop

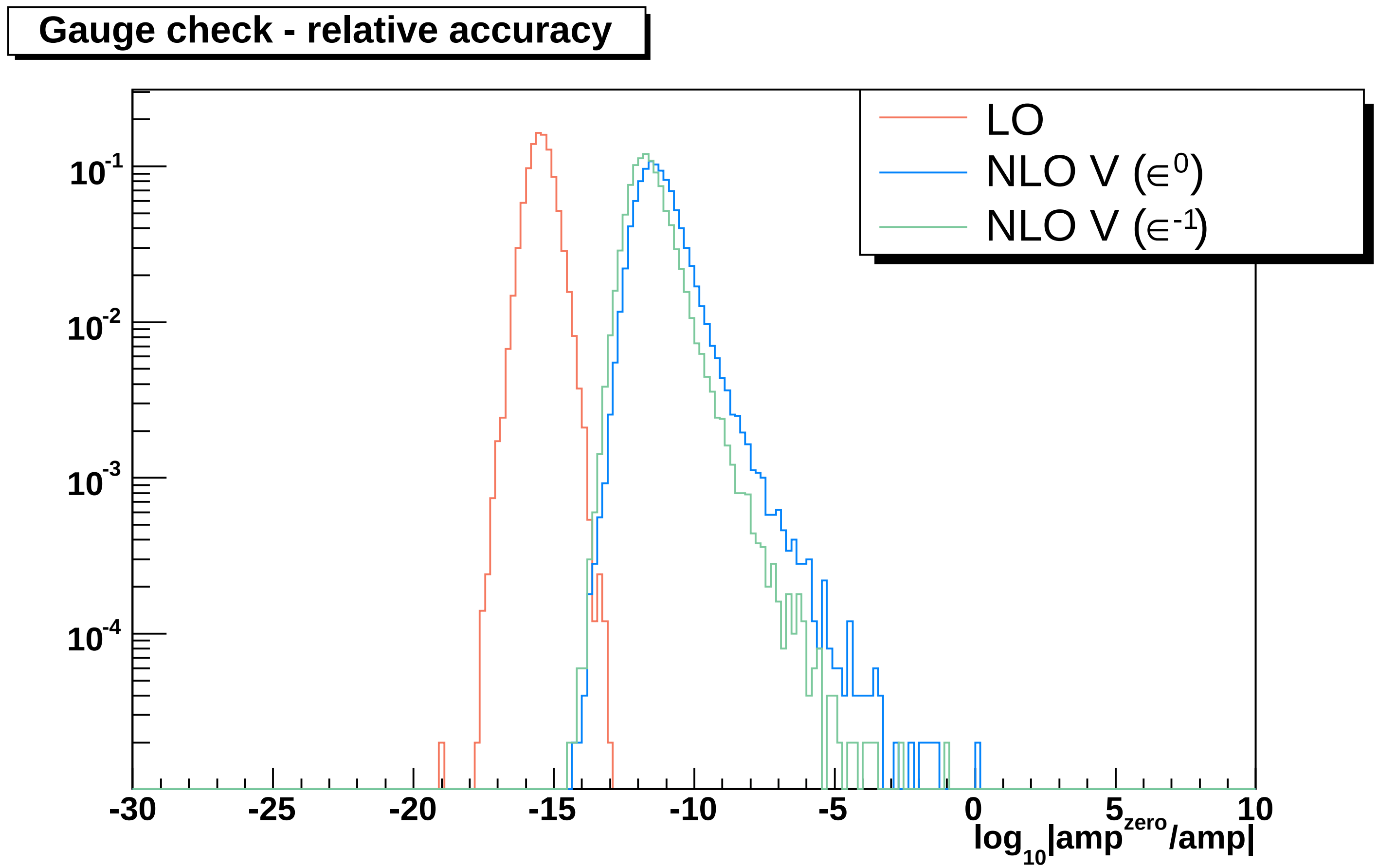
- evaluation one-loop scalar integrals with OneLOop (AvH);
- identification of scalar integrals and calculation of their coefficients as well as R_1 with CutTools (Pittau,Ossola,Garzelli);
- evaluation of the numerator with tree-level amplitude calculator Helac (Papadopoulos,Kanaki,Worek);

Together with the Helac/Phegas system for full cross section calculation (Cafarella,Papadopoulos,Worek,Bevilacqua), it is able to give the virtual contribution in NLO calculations for essentially any process involving up to, at least, 6 particles.

Summary

- NLO precision is needed for LHC;
- preferably obtained with the help of automatic tools;
- based on the OPP method, Helac-1Loop is able to calculate one-loop amplitudes, necessary for the virtual contribution, up to at least 6 particles, eg $\mathrm{pp} \to \mathrm{t\bar{t}\,b\bar{b}}$, $\mathrm{pp} \to W^+W^-\,\mathrm{b\bar{b}}$, $\mathrm{pp} \to \mathrm{b\bar{b}b\bar{b}}$, $\mathrm{pp} \to V\mathrm{ggg}$, $\mathrm{pp} \to \mathrm{t\bar{t}gg}$.
- in combination with Helac-Dipoles (Czakon,Worek), which deals with the dipole subtraction and the real-radiation contribution, it has proven to perform well in full NLO calculations for $\mathrm{pp} \to \mathrm{t\bar{t}\,b\bar{b}}$ (Bevilacqua,Czakon,Papadopoulos,Pittau,Worek).

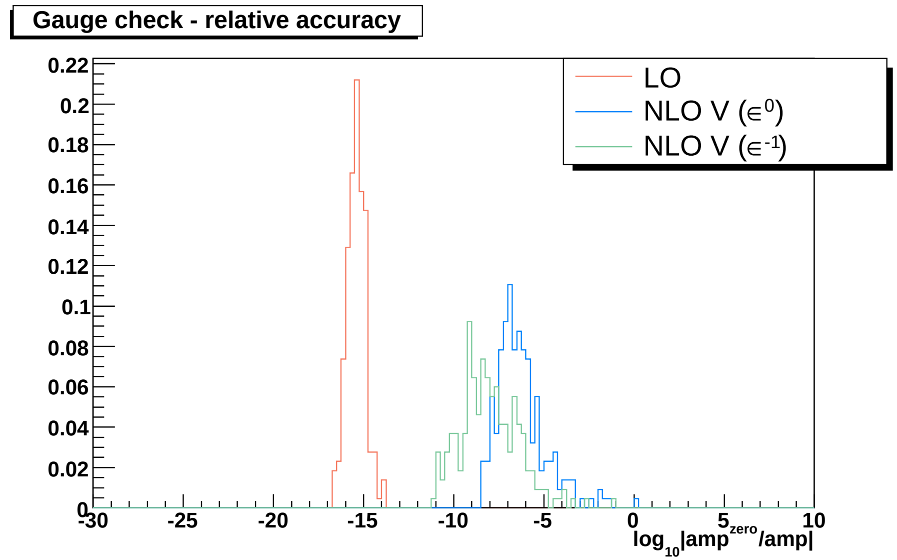
Accuracy: Ward-identity for $gg \to t\bar{t}b\bar{b}$



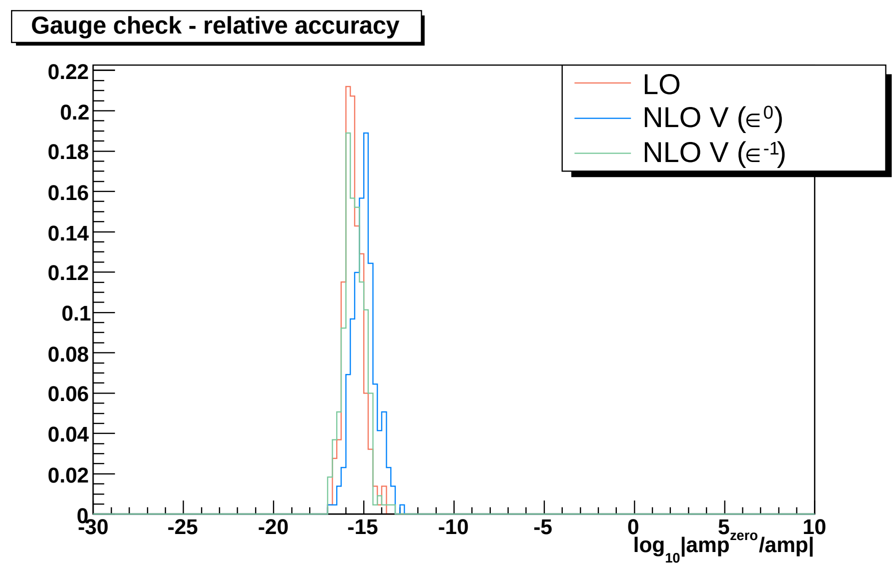
$$\log_{10}\left|\frac{\mathcal{M}(\varepsilon^\mu \leftarrow k^\mu)}{k^0\,\mathcal{M}}\right|$$

Accuracy: Ward-identity for $gg \to t\bar{t}b\bar{b}$

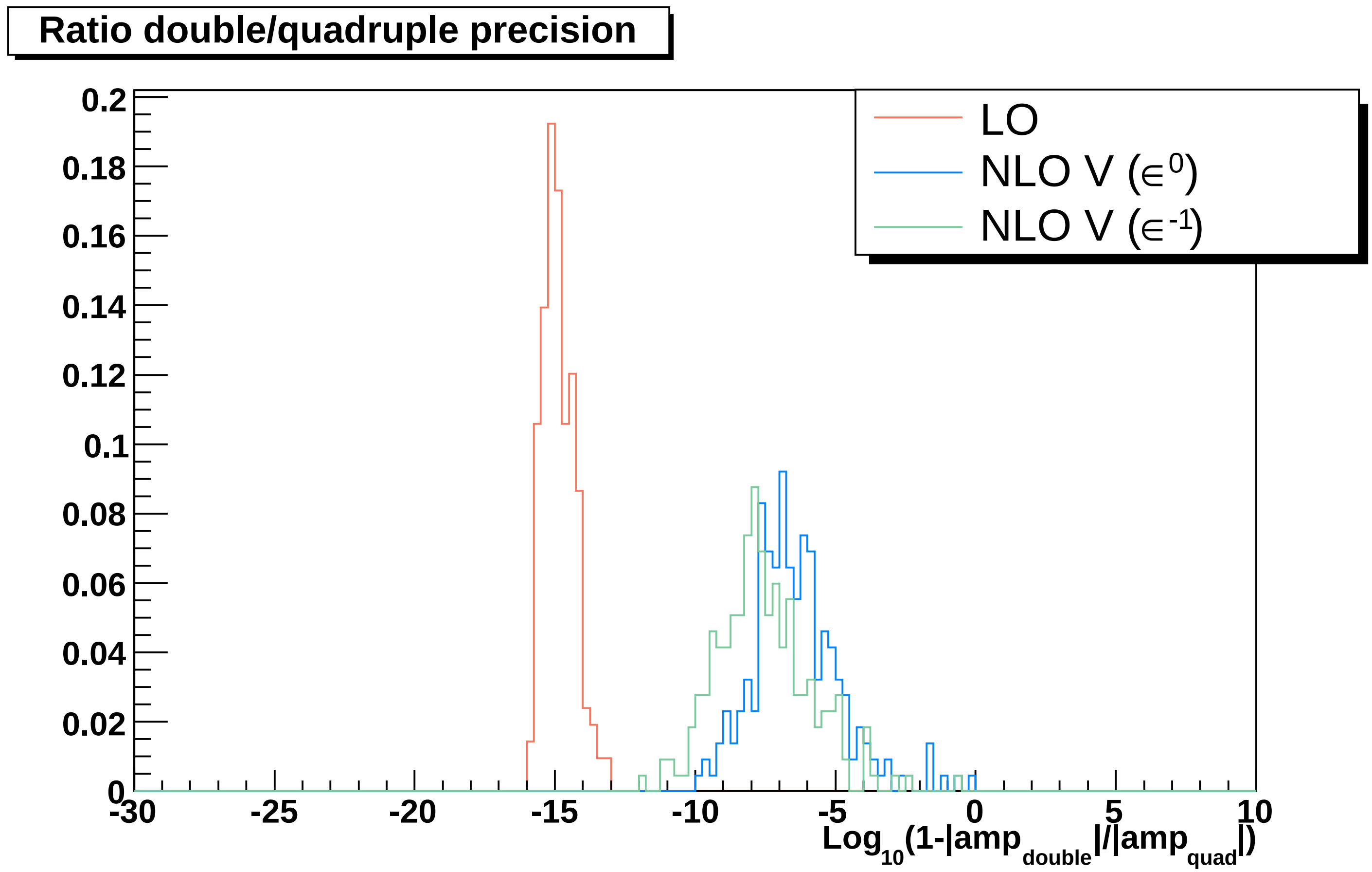
$$\log_{10}\left|\frac{\mathcal{M}(\varepsilon^\mu \leftarrow k^\mu)}{k^0}\right| > -9$$



quadruple precision



Accuracy: Ward-identity for $gg \to t\bar{t}b\bar{b}$



$$\log_{10}\left|\frac{\mathcal{M}(\text{double})}{\mathcal{M}(\text{quadruple})}\right|$$

One-loop amplitude with OPP

For all q:

$$N(q) = \sum_{i_1,i_2,i_3,i_4} \left[\, d(i_1,i_2,i_3,i_4) + \widetilde{d}(q;i_1,i_2,i_3,i_4) \,\right] \prod_{j\neq i_1,i_2,i_3,i_4} D_j$$

$$+ \sum_{i_1,i_2,i_3} \left[\, c(i_1,i_2,i_3) + \widetilde{c}(q;i_1,i_2,i_3) \,\right] \prod_{j\neq i_1,i_2,i_3} D_j$$

$$+ \sum_{i_1,i_2} \left[\, b(i_1,i_2) + \widetilde{b}(q;i_1,i_2) \,\right] \prod_{j\neq i_1,i_2} D_j$$

$$+ \sum_{i} \left[\, a(i) + \widetilde{a}(q;i) \,\right] \prod_{j\neq i} D_j \qquad D_j = (q+p_j)^2 - m_j^2$$

One-loop amplitude with OPP

For all q:

$$N(q) = \sum_{i_1,i_2,i_3,i_4} \left[\, d(i_1,i_2,i_3,i_4) + \widetilde{d}(q;i_1,i_2,i_3,i_4)\,\right] \prod_{j\neq i_1,i_2,i_3,i_4} D_j$$
$$+ \sum_{i_1,i_2,i_3} \left[\, c(i_1,i_2,i_3) + \widetilde{c}(q;i_1,i_2,i_3)\,\right] \prod_{j\neq i_1,i_2,i_3} D_j$$
$$+ \sum_{i_1,i_2} \left[\, b(i_1,i_2) + \widetilde{b}(q;i_1,i_2)\,\right] \prod_{j\neq i_1,i_2} D_j$$
$$+ \sum_{i} \left[\, a(i) + \widetilde{a}(q;i)\,\right] \prod_{j\neq i} D_j \qquad D_j = (q+p_j)^2 - m_j^2$$

Choose $q = q_0$ such that $D_{i_1} = D_{i_2} = D_{i_3} = D_{i_4} = 0$:

$$N(q_0) = \left[\, d(i_1,i_2,i_3,i_4) + \widetilde{d}(q_0;i_1,i_2,i_3,i_4)\,\right] \prod_{j\neq i_1,i_2,i_3,i_4} D_j$$

There are exactly 2 such q_0, enough to determine $d, \widetilde{d}$. So by using values of q such that denominators are zero, the equation triangularizes.