# Next-to-Leading Order: MadLoop & MadFKS

## Lecture I


**Rikkert Frederix**

**University of Zurich**

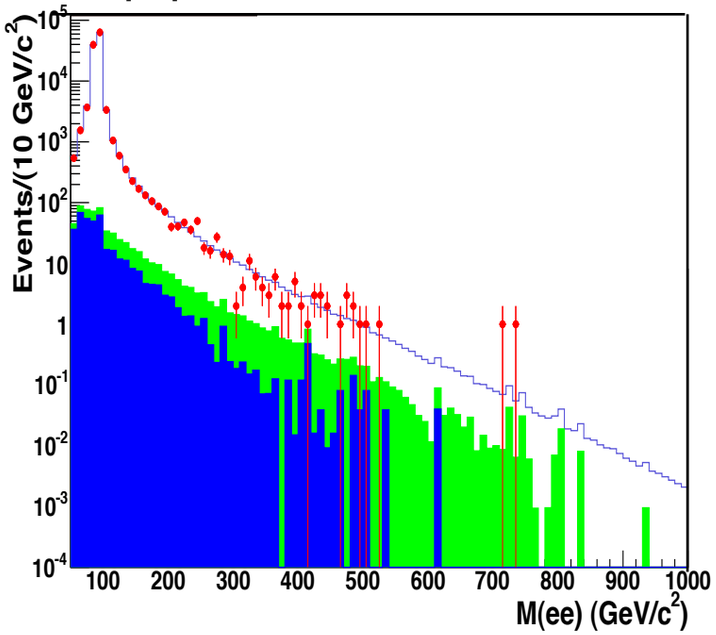


Think Tank on Physics@LHC: Monte Carlo Event Generators at NLO and Jet Physics, Sariska Palace, Rajasthan, India, Dec. 5-9, 2011

# Discoveries at hadron colliders

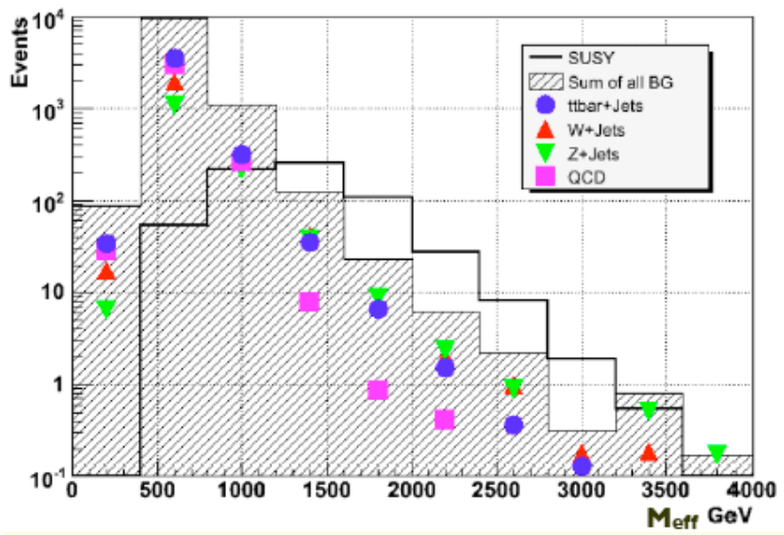
## peak

$pp \to Z' \to e^+e^-$



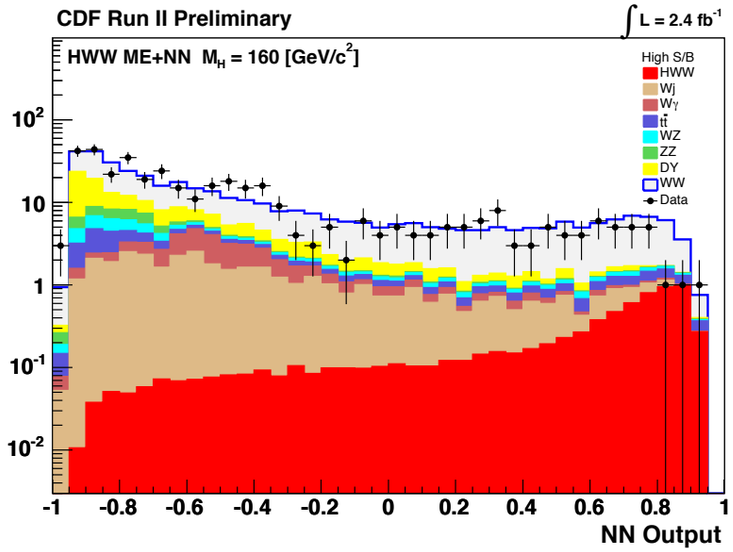

## shape

$pp \to \tilde{g}\tilde{g}, \tilde{g}\tilde{q}, \tilde{q}\tilde{q} \to \text{jets} + \not{E}_T$




## rate

$pp \to H \to W^+W^-$




## "easy"

Background directly measured from data. TH needed only for parameter extraction (Normalization, acceptance,...)

## hard

Background shapes needed. Flexible MC for both signal and backgroud tuned and validated with data.

## very hard

Background normalization and shapes known very well. Interplay with the best theoretical predictions (via MC) and data.

# Leading order

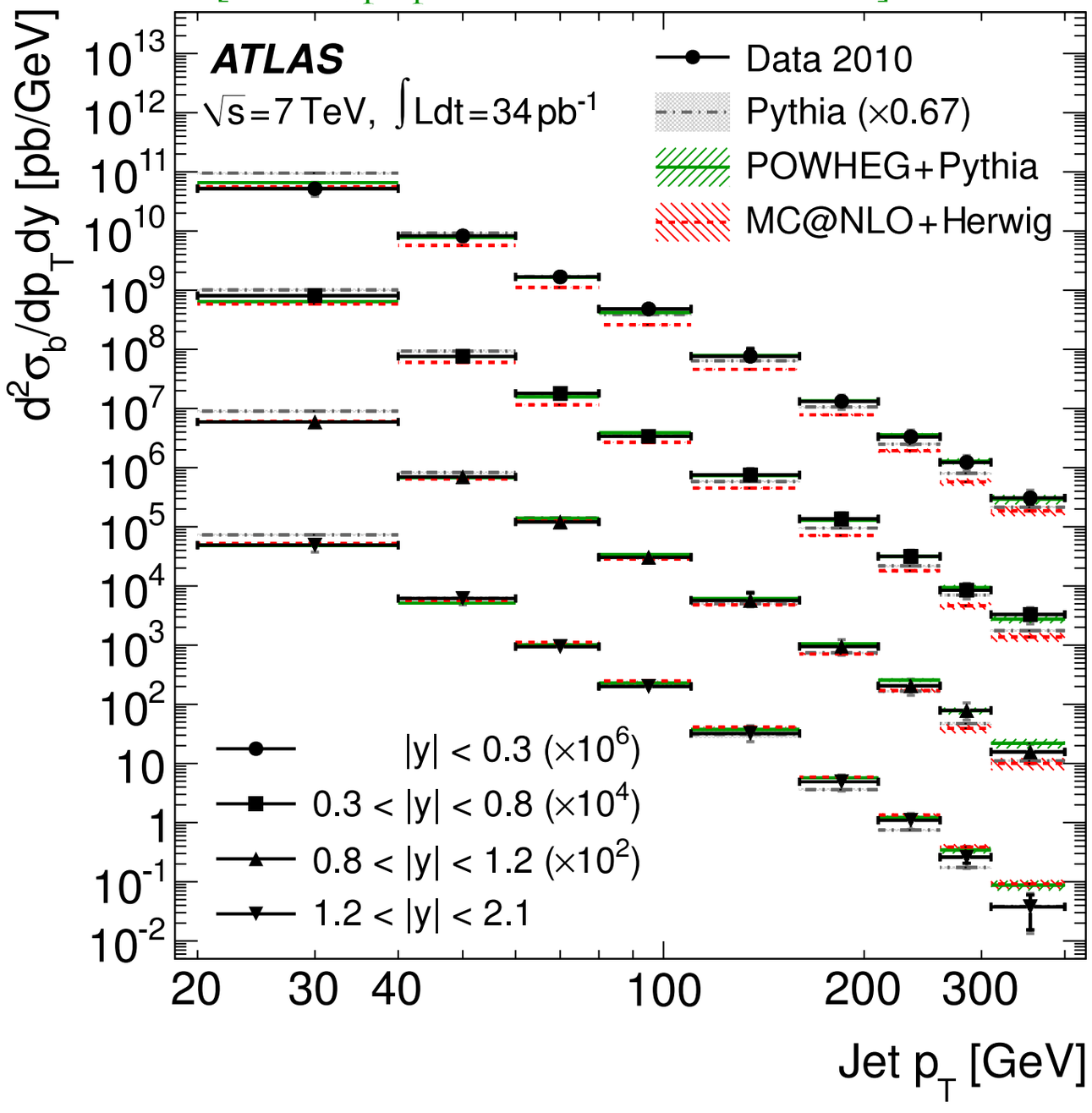
- For many of the theory predictions needed in the searches for new physics as well as measuring properties of the SM, leading order predictions are used
  - In many regions of phase-space they do a (surprisingly) good job, in particular for shapes of distributions
  - Flexible computer codes readily available and relatively easy to use and understand
- Unfortunately LO predictions describe total rates rather poorly: you only get the order of magnitude

# Need for NLO

- If we would have the same **flexible** tools available at NLO, analyses will benefit a various ways:
  - NLO predictions predict rates much more precisely
  - Reduced theoretical uncertainties due to meaningful scale dependence
  - Shapes of distributions are better described
  - Correct estimates for PDF uncertainties
- These accurate theoretical predictions are particularly needed for
  - searches of signal events in large backgrounds samples and
  - precise extraction of parameters (couplings etc.) when new physics signals have been found

# B pair production at the LHC

[ATLAS, preprint CERN-TH-EP-2011-146]




- b-jet transverse momentum in b pair production
- LO: Pythia,
  NLO: POWHEG + Pythia
  NLO: MC@NLO + Herwig
- Shapes are well described by LO and by NLO
- Normalization is well predicted at NLO
- At NLO, theory uncertainties can be studied systematically (not shown in this plot)

# Why an automatic tool?

- To save time

  Less human time spending on computing matrix elements means more time available on physics and phenomenology.

- Robustness

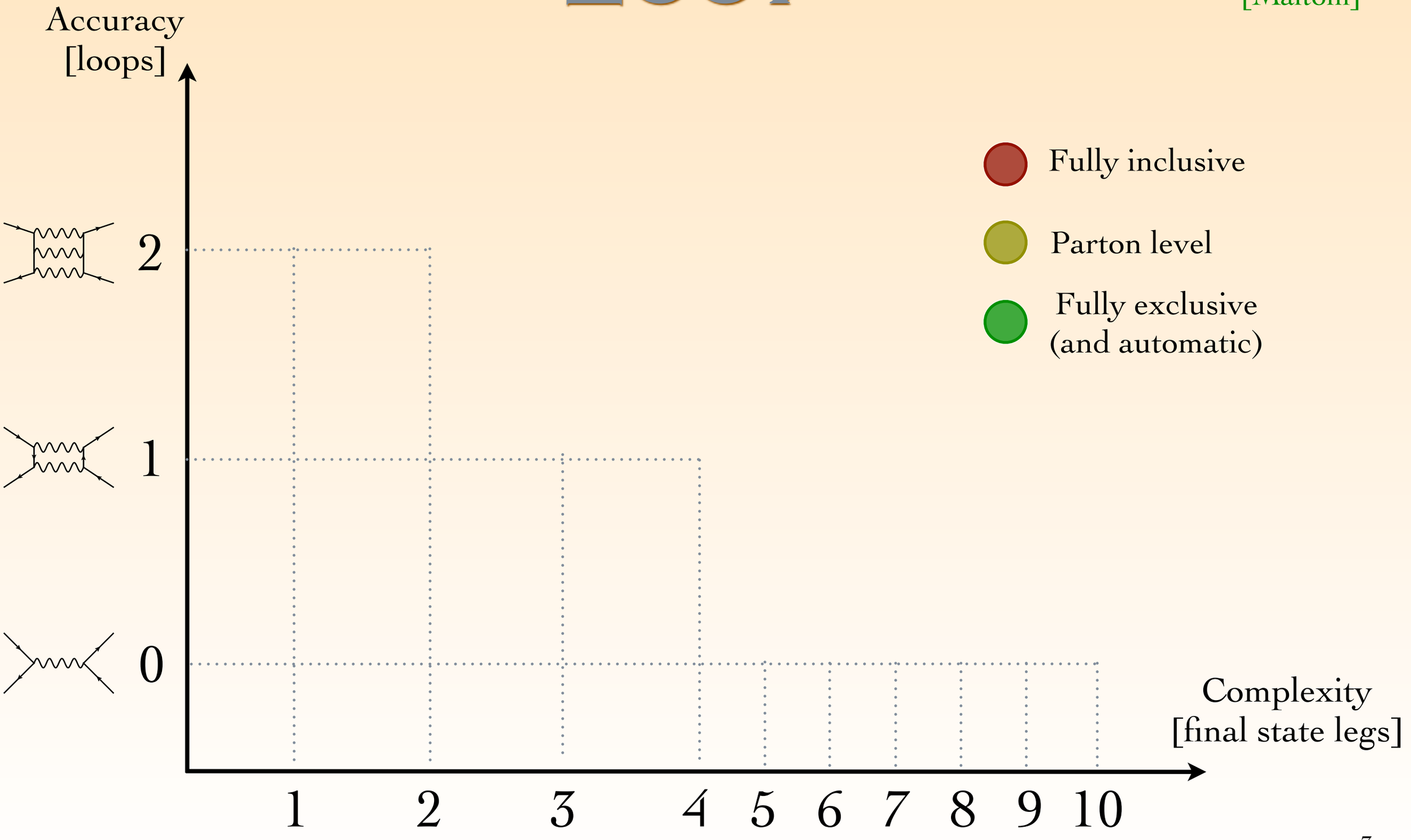
  Modular code structure means that elements can be checked systematically and extensively once and for all. Trust can easily be build.

- Wide accessibility

  One framework for all. Available to everybody for an unlimited set of applications. Suitable for Experimental collaborations.

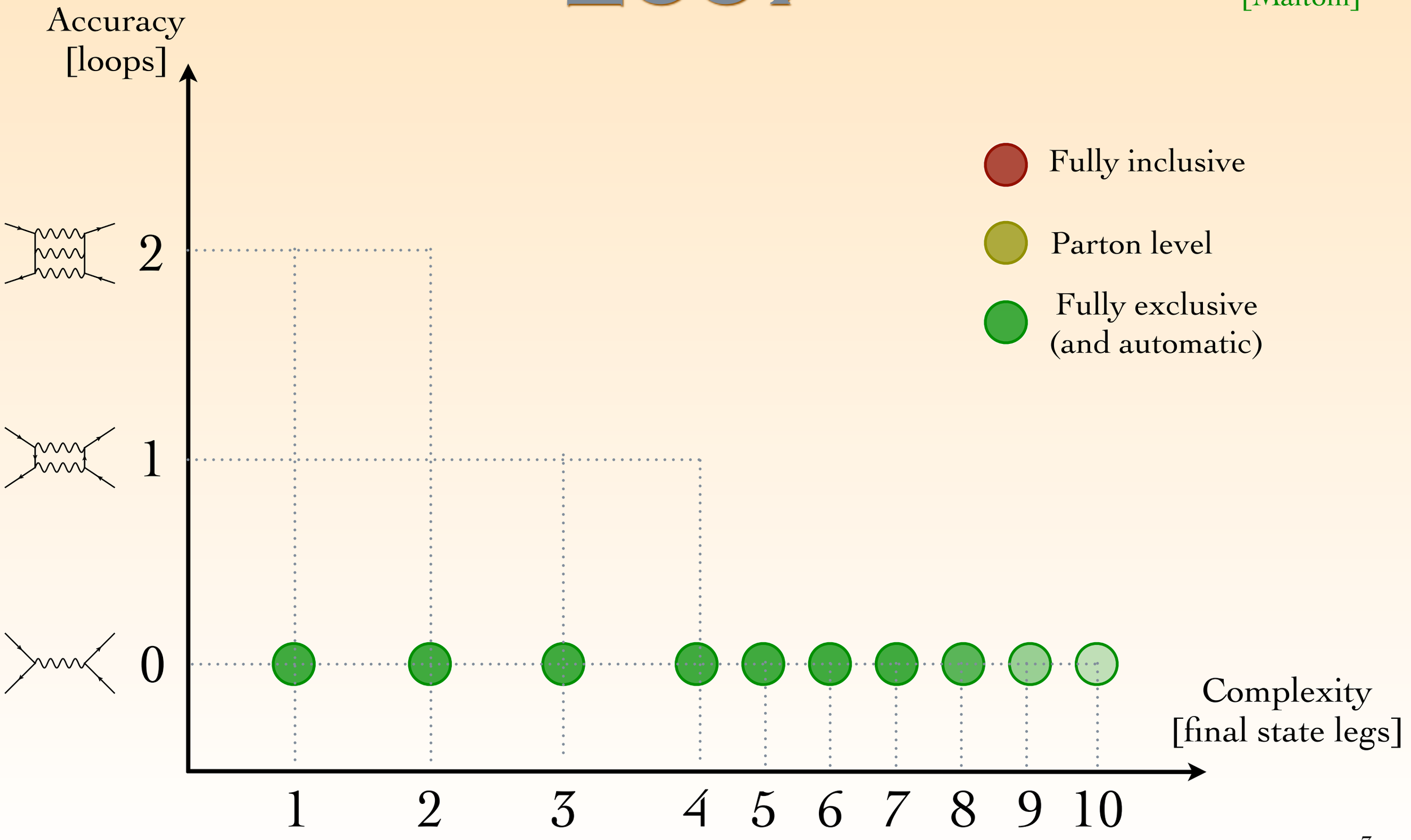
# SM status: since 2007

[Maltoni]



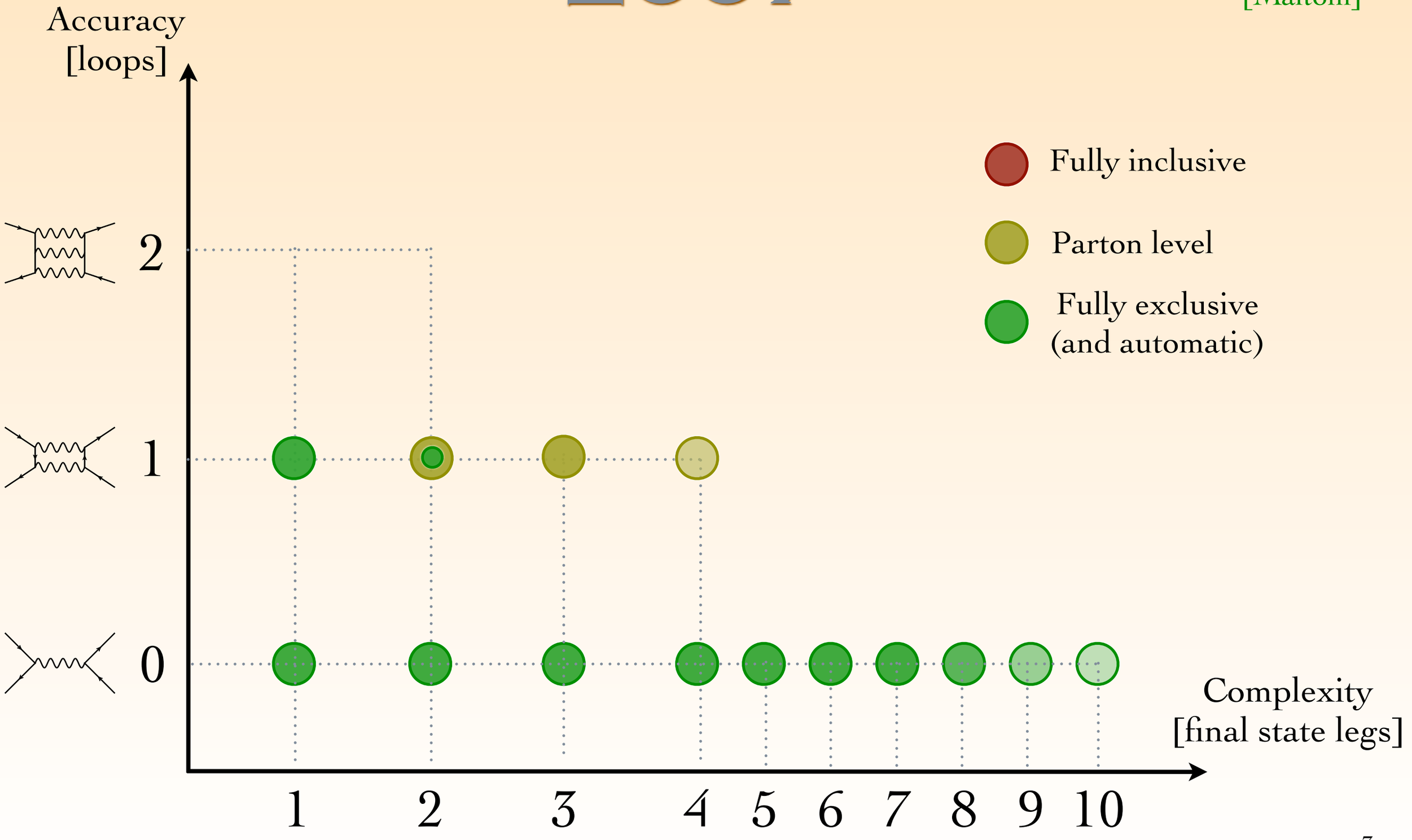
# SM status: since 2007

[Maltoni]



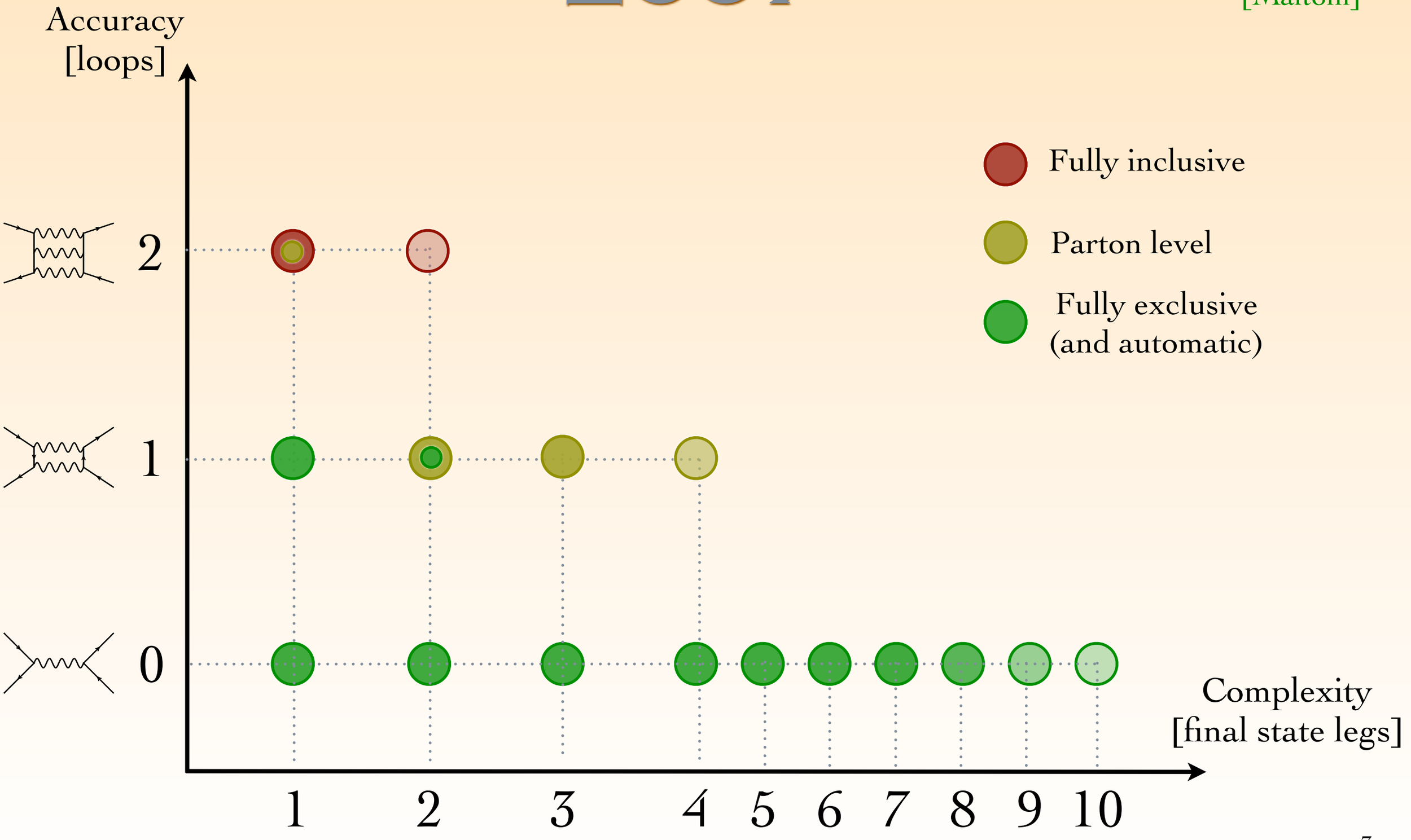
# SM status: since 2007

[Maltoni]

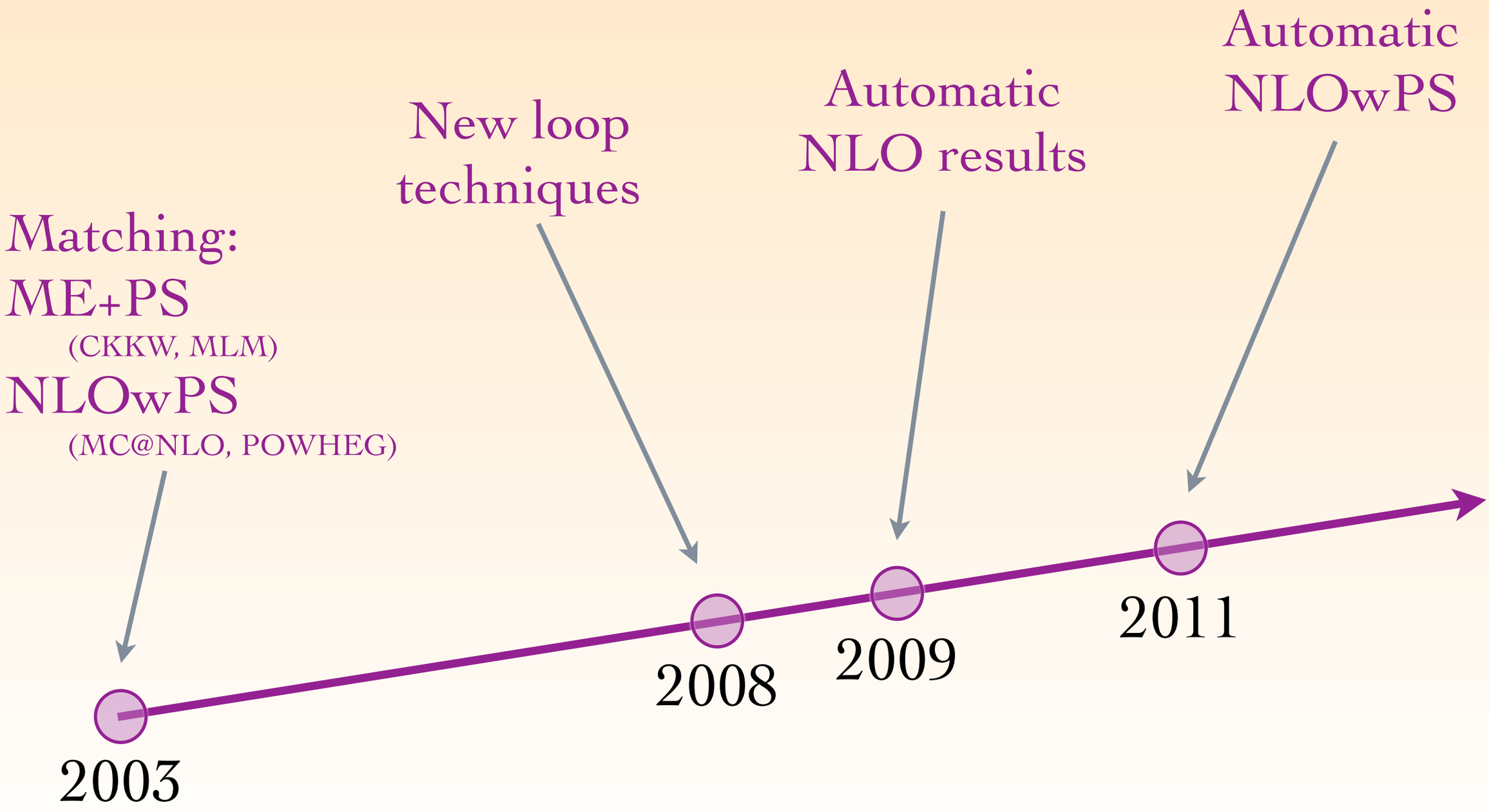


# SM status: since 2007

[Maltoni]

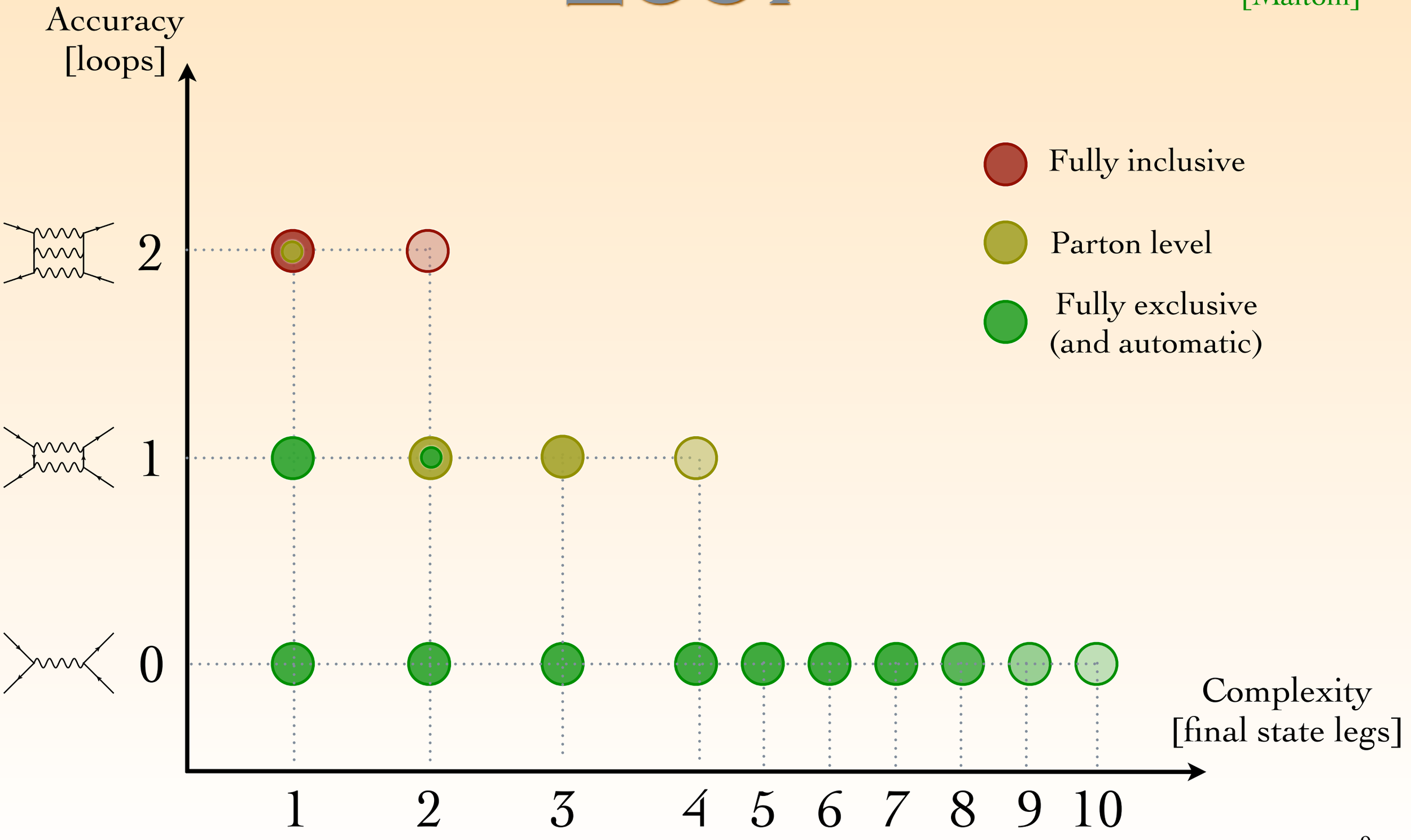


# QCD and MC progress (simplified)



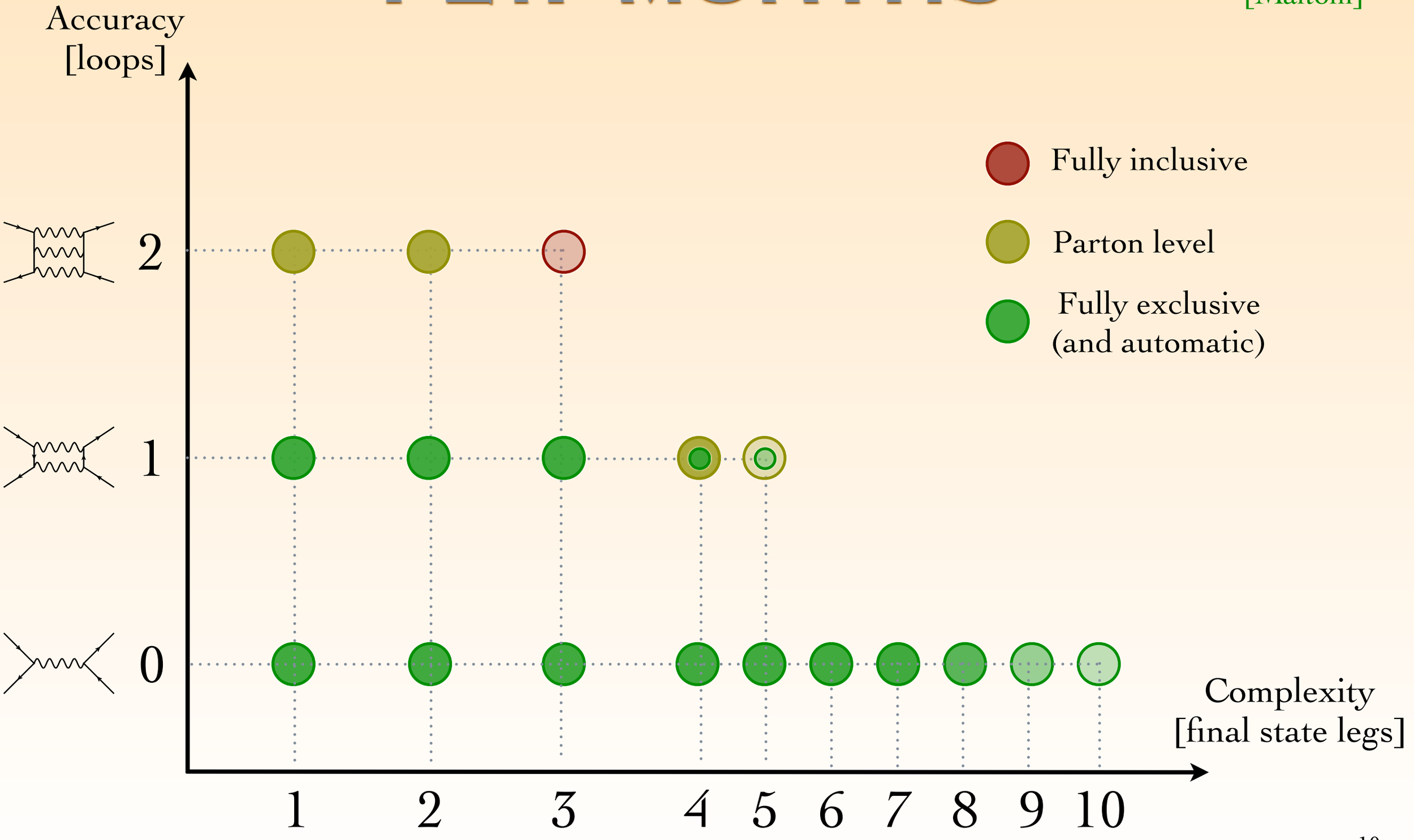
# SM status: since 2007

[Maltoni]



# SM status: since a few months

[Maltoni]



# Contents of these lectures

- These lectures will be about the concepts behind the computation of NLO corrections (as implemented in MadGraph)
  - Cancellation of infrared singularities
    - FKS subtraction in MadFKS
  - Computing Loop corrections without doing integrals and using only tree-level matrix elements
    - MadLoop
  - Tutorial
- Please, interrupt me at any time if something is not clear!

# Not yet publicly available

- Disclaimer:
  Although NLO corrections are included in MadGraph (version 4) they are not yet publicly available. The reason is that the code is
  - still a bit slow (not possible to run on a single desktop anything beyond a $2 \to 1$ process within 30 minutes or so)
  - not yet completely general (some processes cannot be computed; and no warning is given if one tries)
- The code is being rewritten in MG5. When this is done, it will be made publicly available
- The tutorial will be based on parts of the automatic tool: MadFKS with virtual matrix elements taken from MCFM
  [J.M. Campbell & R.K. Ellis; + C. Williams; + external collaborators]

# NLO computations

# Master equation for hadron colliders

$$d\sigma = \sum_{a,b} \int dx_1 dx_2 \; f_a(x_1, \mu_F) f_b(x_2, \mu_F) \, d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$$

Parton density functions

Parton-level (differential) cross section

- Parton-level cross section from matrix elements: model and process dependent
- Parton density (or distribution) functions: process independent
- Differences between colliders given by parton luminosities

# Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

- Including higher corrections improves predictions and reduces theoretical uncertainties

# Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

- Including higher corrections improves predictions and reduces theoretical uncertainties

# Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

# Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

NNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

# Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

NNLO corrections

NNNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

# Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

NNLO corrections

NNNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

# NLO CORRECTIONS

- There are two types of contributions to the NLO corrections:
  - Virtual (or Loop) corrections: formed by an amplitude with a closed loop of particles interfered with the Born amplitudes
  - Real emission corrections: formed by amplitudes with one extra parton compared to the Born process
- Both have one power of $\alpha_s$ extra compared to the Born process

$$\sigma^{\mathrm{NLO}} = \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(d)}\sigma^V + \int_m d^{(4)}\sigma^B$$

# NLO PREDICTIONS

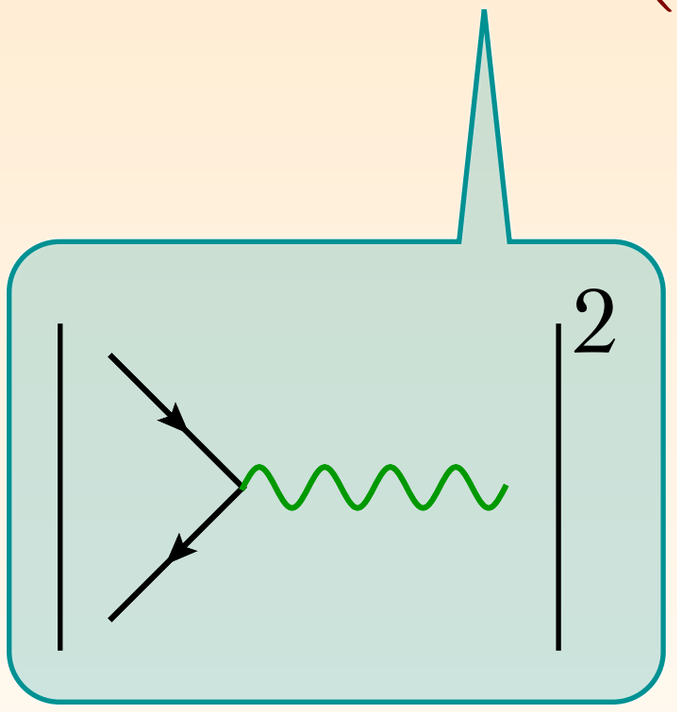
- As an example, consider Drell-Yan production

$$\hat{\sigma} = \sigma^{\text{Born}} \left( 1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \dots \right)$$

# NLO PREDICTIONS

- As an example, consider Drell-Yan production

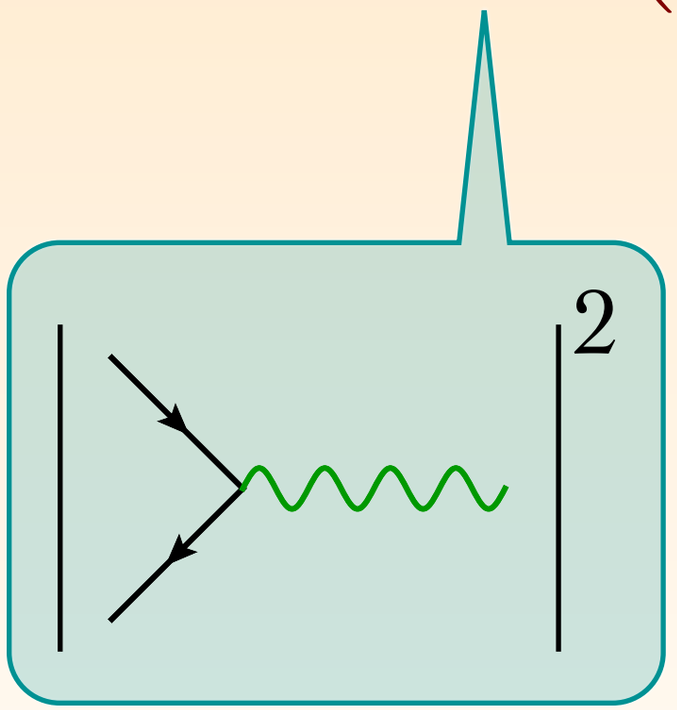
$$\hat{\sigma} = \sigma^{\text{Born}} \left( 1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \dots \right)$$

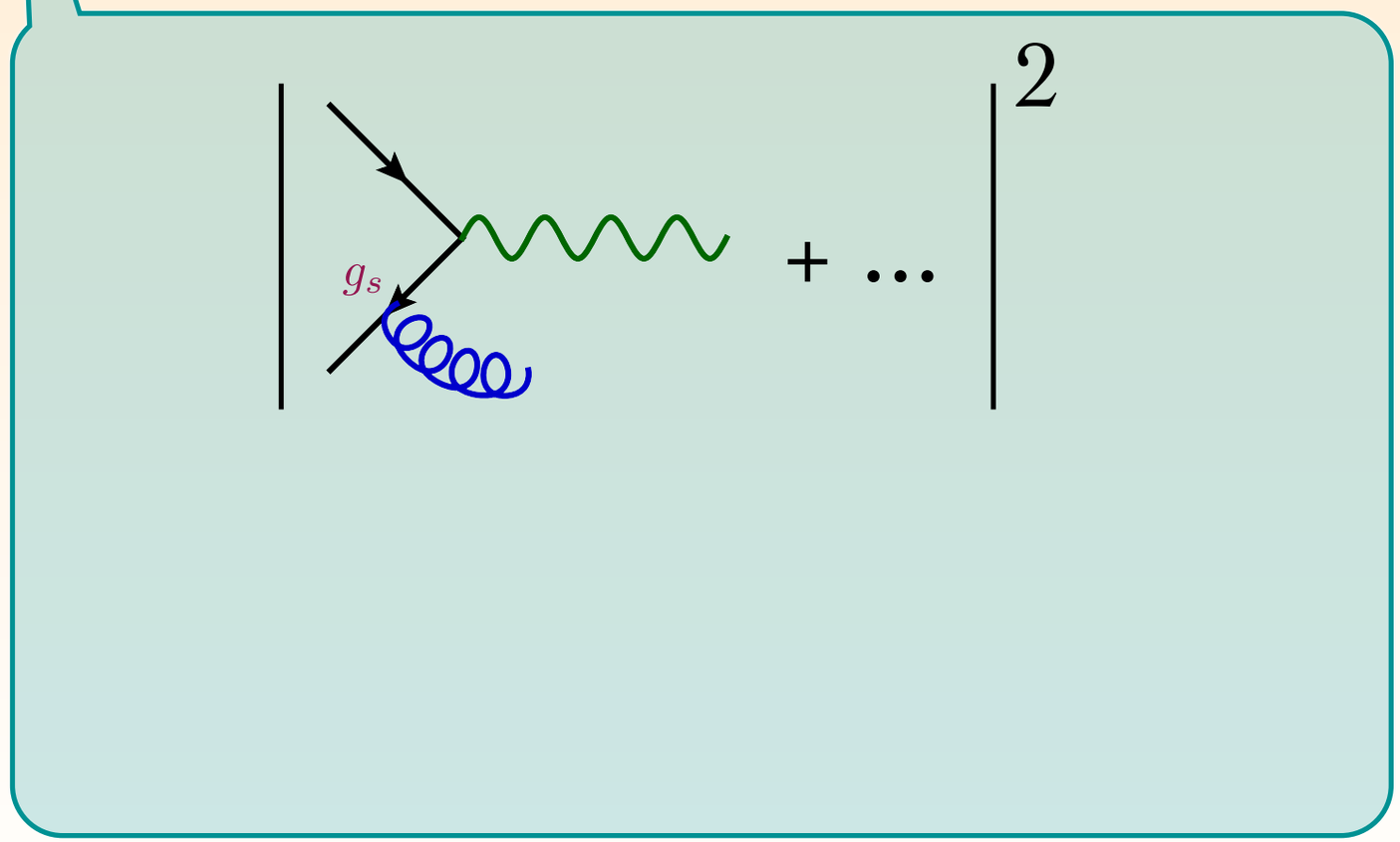


# NLO PREDICTIONS

- As an example, consider Drell-Yan production

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \ldots\right)$$

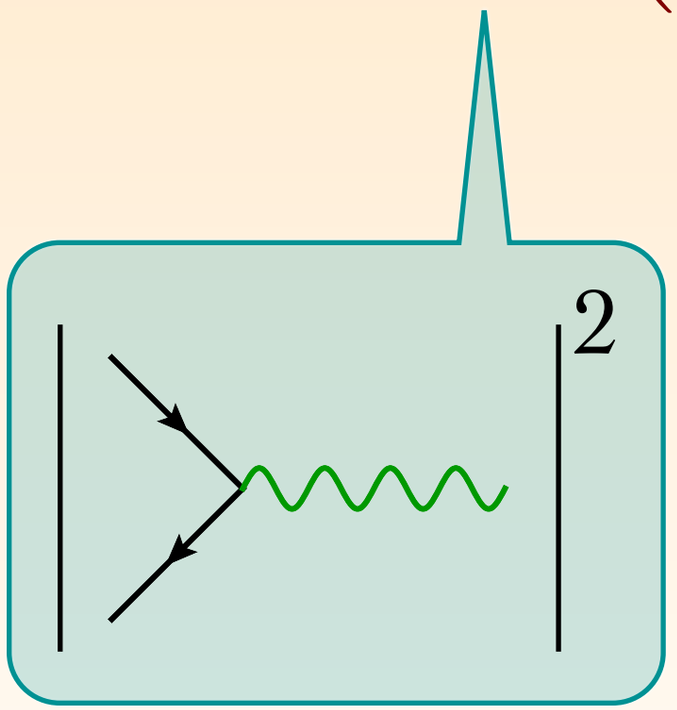


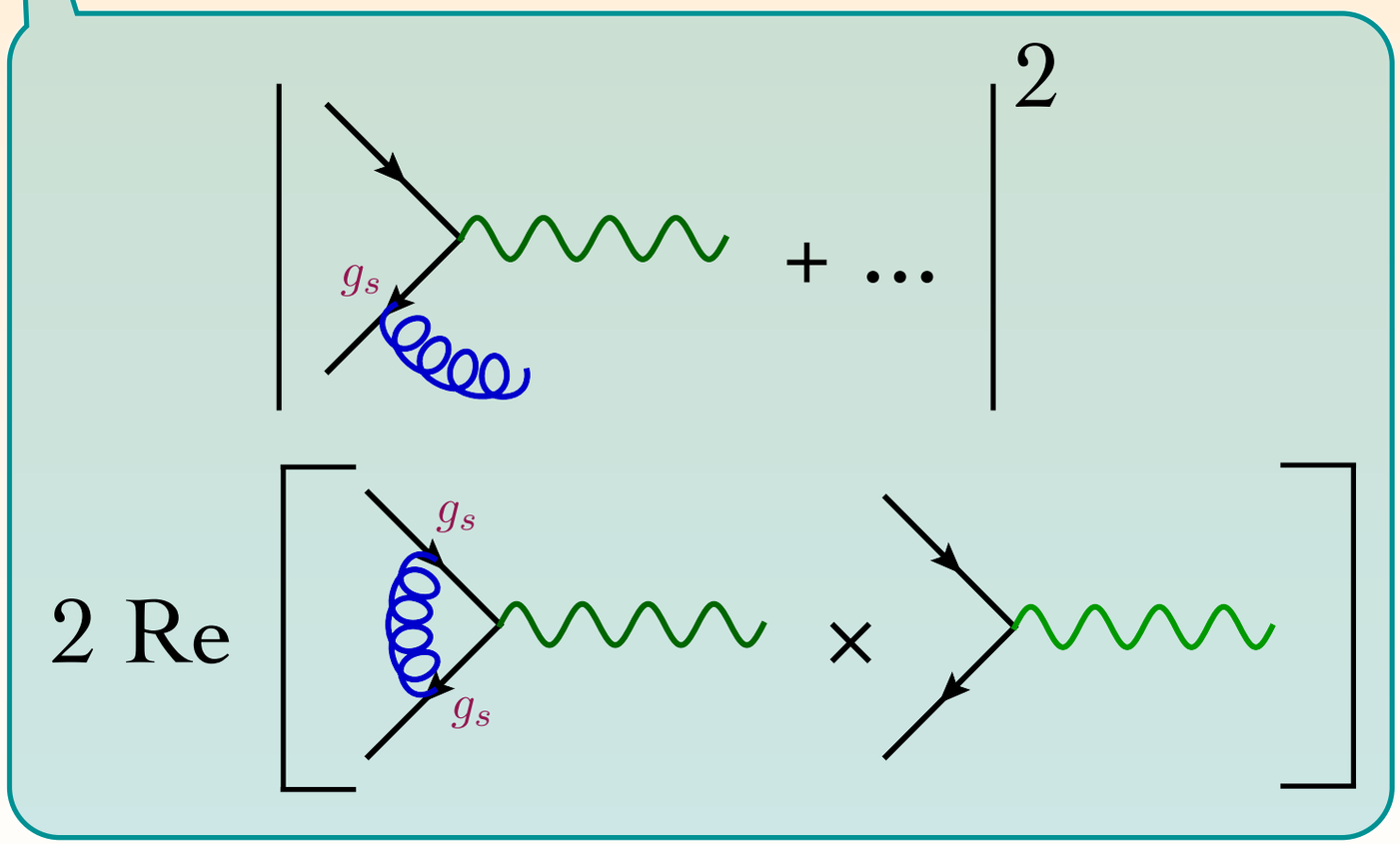



# NLO PREDICTIONS

- As an example, consider Drell-Yan production

$$\hat{\sigma} = \sigma^{\text{Born}} \left( 1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \ldots \right)$$





# Improved predictions

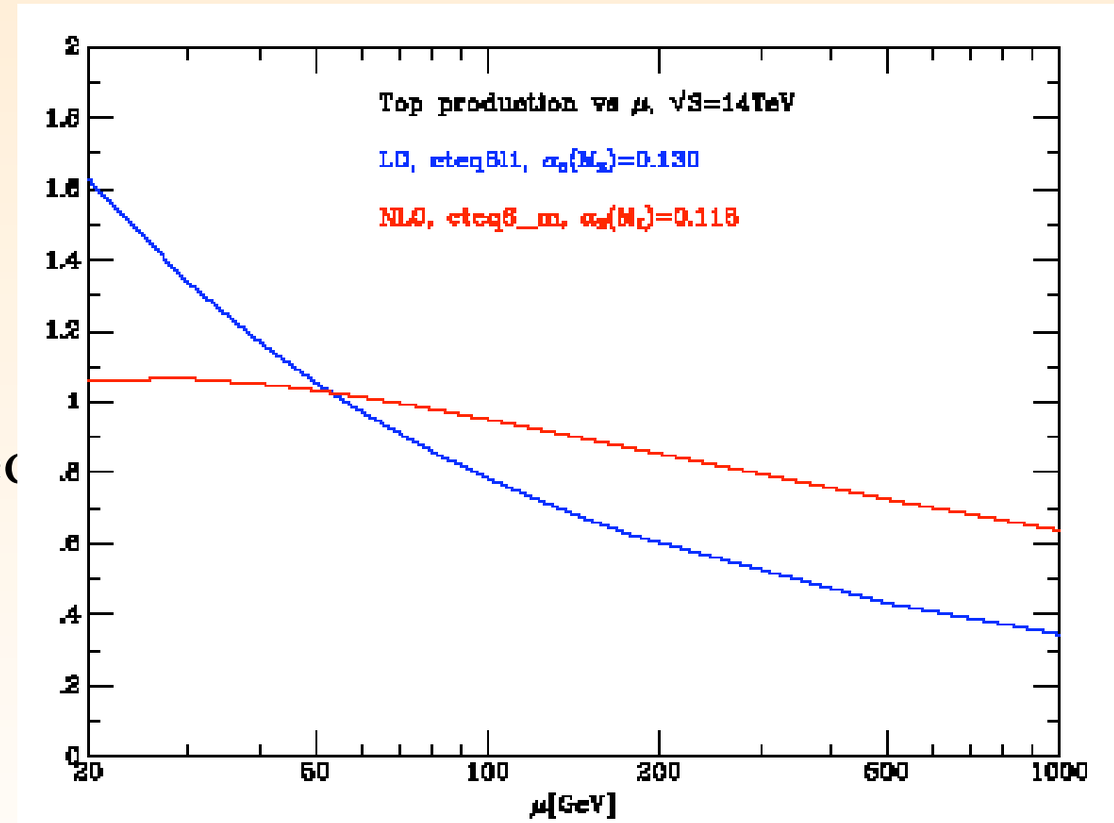
$$d\sigma = \sum_{a,b} \int dx_1 dx_2 \; f_a(x_1, \mu_F) f_b(x_2, \mu_F) \, d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$$

$$\hat{\sigma} = \sigma^{\text{Born}} \left( 1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \left( \frac{\alpha_s}{2\pi} \right)^2 \sigma^{(2)} + \left( \frac{\alpha_s}{2\pi} \right)^3 \sigma^{(3)} + \ldots \right)$$

- Remember, predictions are inclusive: also at LO initial state radiation is included via the PDF; final state radiation by the definition of the parton, which represents all final state evolutions
- Due to these approximations, Leading Order predictions can depend strongly on the renormalization and factorization scales
- Including higher order corrections reduces the dependence on these scales

# Going NLO

- At NLO the dependence on the renormalization and factorization scales is reduced
  - First order scale dependence in the running coupling and the PDFs is compensated for via the loop corrections
  - Better description of final state: impact of extra radiation included (e.g. jets can have substructure)
  - Opening of additional initial state partonic channels



# NLO…?

- Are all (IR-safe) observables that we can compute using a NLO calculation correctly described at NLO?

- It depends on the observable…
- In the small transverse momentum region, this calculation breaks down (it's even negative in the first bin!), and anywhere else it is purely a LO calculation for V+1j

# NLO...?

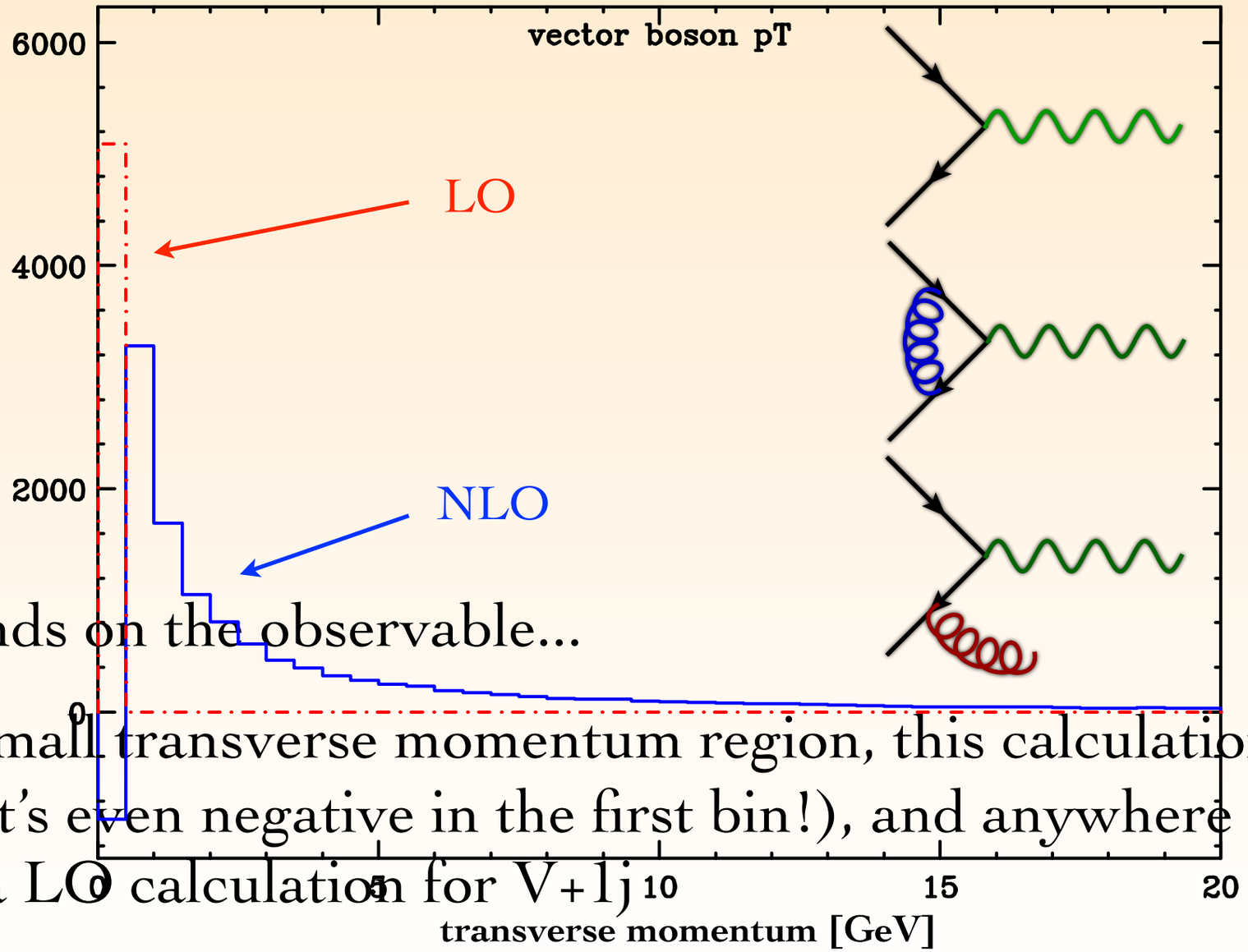
- Are all (IR-safe) observables that we can compute using a NLO calculation correctly described at NLO?

**NO!**

- It depends on the observable...
- In the small transverse momentum region, this calculation breaks down (it's even negative in the first bin!), and anywhere else it is purely a LO calculation for V+1j

# NLO…?

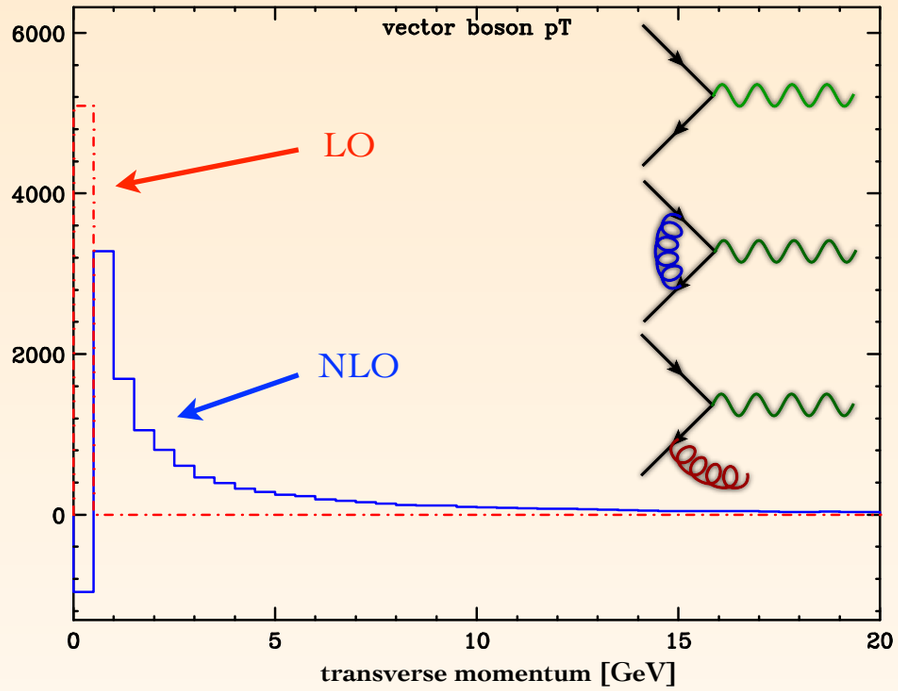
- Are all (IR-safe) observables that we can compute using a NLO calculation correctly described at NLO?




- It depends on the observable…
- In the small transverse momentum region, this calculation breaks down (it's even negative in the first bin!), and anywhere else it is purely a LO calculation for V+1j

# NLO…?

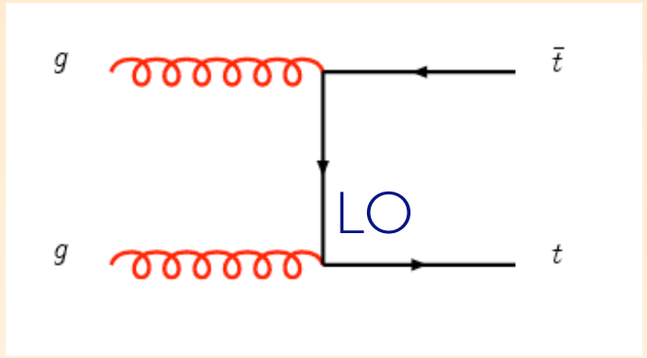
- Are all (IR-safe) observables that we can compute using a NLO calculation correctly described at NLO?

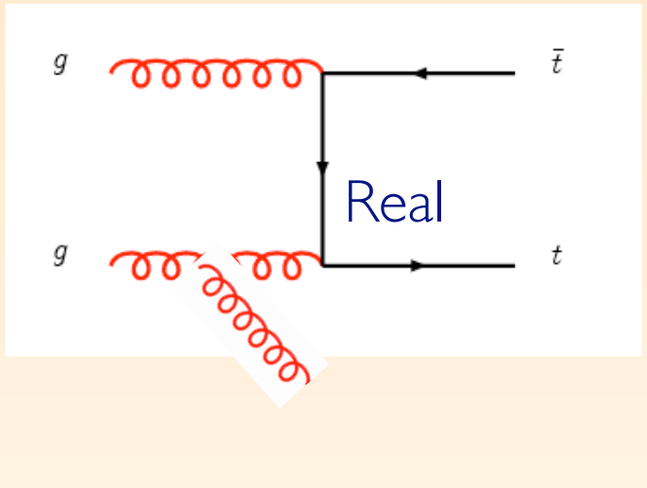


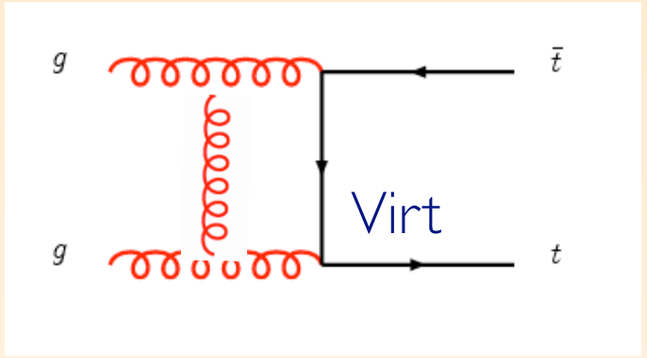

- It depends on the observable…
- In the small transverse momentum region, this calculation breaks down (it's even negative in the first bin!), and anywhere else it is purely a LO calculation for V+1j

# NLO…?

- Another example: we have a NLO code for pp → tt

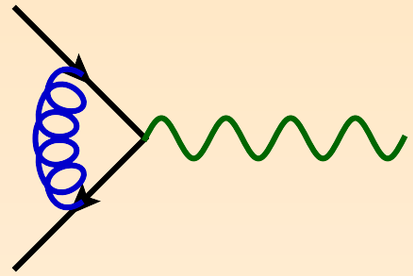



| | NLO? |
|---|---|
| Total cross section | ✔ |
| Transverse momentum of the top quark | ✔ |
| Transverse momentum of the top-antitop pair | ✘ |
| Transverse momentum of the jet | ✘ |
| Top-antitop invariant mass | ✔ |
| Azimuthal distance between the top and anti-top | ✘ |

# Bottlenecks



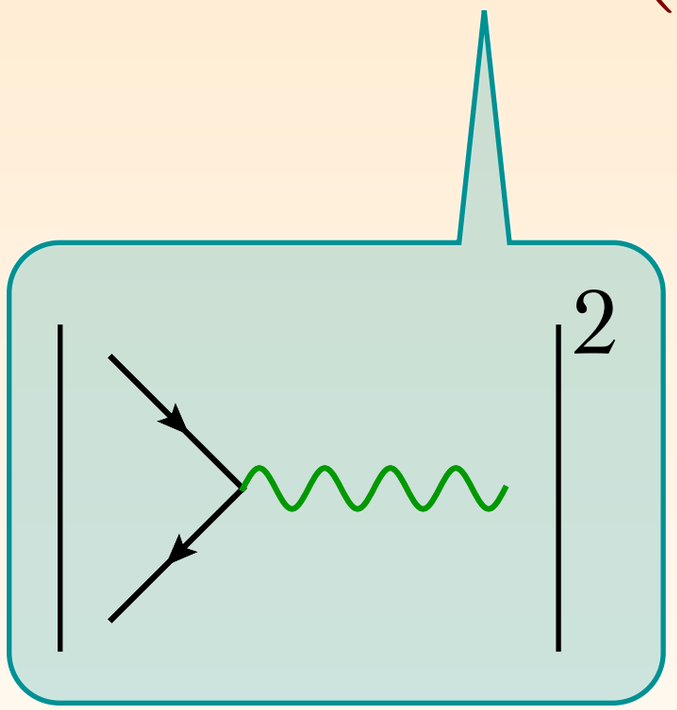
- Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time
- How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem
- How to match these processes to a parton shower without double counting (will be covered by Paolo Torrielli's lectures)

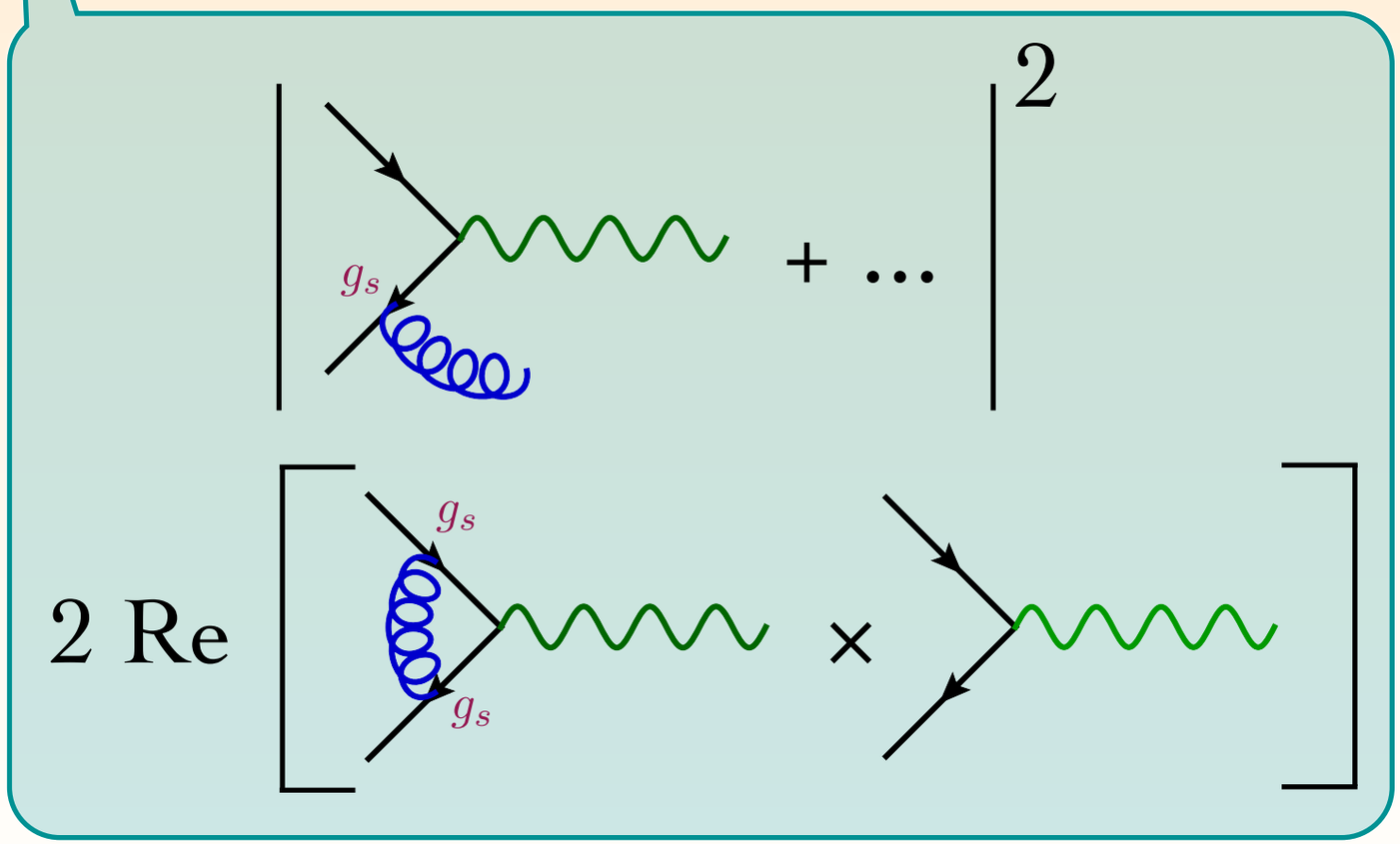
# Canceling infrared divergences: FKS subtraction

# NLO PREDICTIONS

- As an example, consider Drell-Yan production

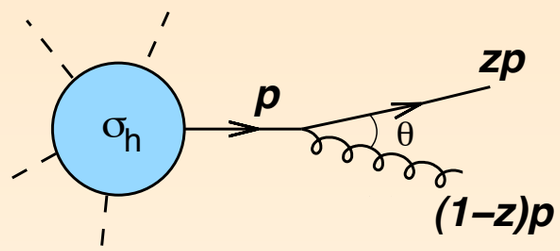
$$\hat{\sigma} = \sigma^{\text{Born}} \left( 1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \ldots \right)$$





# Branching

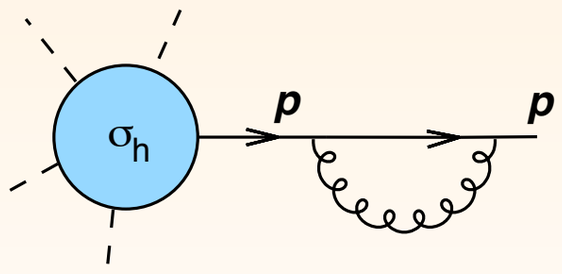
- In the soft and collinear region, the branching of a gluon from a quark can be written as




$$\sigma_{h+g} \simeq \sigma_h \frac{\alpha_s C_F}{\pi} \frac{dz}{1-z} \frac{dk_t^2}{k_t^2}$$

where $k_t$ is the transverse momentum of the gluon, $k_t = E \sin\theta$.

- The singularities cancel against the singularities in the virtual corrections, which result from the integral over the loop momentum of the function




$$\sigma_{h+V} \simeq -\sigma_h \frac{\alpha_s C_F}{\pi} \frac{dz}{1-z} \frac{dk_t^2}{k_t^2}$$

- The sum is finite for observables that cannot distinguish between two collinear partons ($k_t \longrightarrow 0$); a hard and a soft parton ($z \longrightarrow 1$); and a single parton (in the virtual contributions)

# Infrared cancellation

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m) + \int d^4\Phi_m \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_{m+1} \, R(\Phi_{m+1})$$

- The KLN theorem tells us that divergences from virtual and real-emission corrections cancel in the sum (for observables insensitive to soft and collinear radiation)
- When doing an analytic calculation in dimensional regularization this can be explicitly seen in the cancellation of the $1/\epsilon$ and $1/\epsilon^2$ terms (with $\epsilon$ the regulator, $\epsilon \to 0$)
- In the real emission corrections, the explicit poles enter after the phase-space integration (in d dimensions)

# Infrared safe observables

- For an observable to be calculable in fixed-order perturbation theory, the observable should be infrared safe, i.e., it should be insensitive to the emission of soft or collinear partons.

- In particular, if $p_i$ is a momentum occurring in the definition of an observable, it most be invariant under the branching

  $$p_i \longrightarrow p_j + p_k,$$

  whenever $p_j$ and $p_k$ are collinear or one of them is soft.

- Examples

  - "The number of gluons" produced in a collision is not an infrared safe observable

  - "The number of hard jets defined using the $k_T$ algorithm," produced in a collision is an infrared safe observable

# PHASE-SPACE INTEGRATION

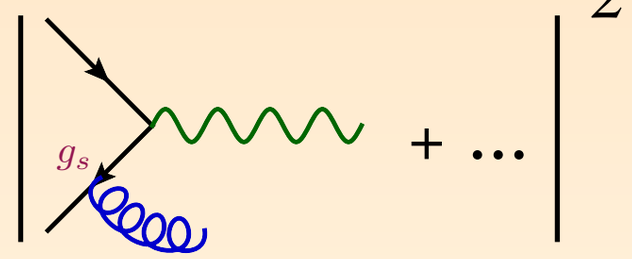
$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m) + \int d^4\Phi_m \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_{m+1} \, R(\Phi_{m+1})$$

- For complicated processes we have to result to numerical phase-space integration techniques ("Monte Carlo integration"), which can only be performed in an integer number of dimensions
  - Cannot use a finite value for the dimensional regulator and take the limit to zero in a numerical code
- But we still have to cancels the divergences explicitly
- Two commonly used solutions exists
  - Phase-space slicing
  - Subtraction method

# Example

- Suppose we want to compute the integral ("real emission radiation", where the 1-particle phase-space is referred to as the 1-dimensional $x$)

$$\int_0^1 dx\, f(x)$$




where $f(x) = \dfrac{g(x)}{x}$ and $g(x)$ is finite everywhere

- Let's introduce a regulator

$$\lim_{\epsilon\to 0}\int_0^1 dx\, \frac{g(x)}{x^{1+\epsilon}} = \lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x)$$

for any non-integer non-zero value for $\epsilon$ this integral is finite

- We would like to factor out the explicit poles in $\epsilon$ so that they can be canceled explicitly against the "virtual corrections"

# Phase-space slicing

$$\lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x) \qquad\qquad f(x)=\frac{g(x)}{x}$$

- Introduce a small parameter $\delta$

$$\begin{aligned}
\lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x) &= \lim_{\epsilon\to 0}\left[\int_0^\delta dx\, x^{-\epsilon} f(x) + \int_\delta^1 dx\, x^{-\epsilon} f(x)\right] \\
&= \lim_{\epsilon\to 0}\left[\int_0^\delta dx\, x^{-\epsilon}\frac{g(0)}{x} + \int_\delta^1 dx\, x^{-\epsilon}\frac{g(x)}{x^{1+\epsilon}}\right] \\
&= \lim_{\epsilon\to 0}\frac{\delta^{-\epsilon}}{-\epsilon} g(0) + \int_\delta^1 dx\,\frac{g(x)}{x} \\
&= \lim_{\epsilon\to 0}\left[\frac{-1}{\epsilon}+\log\delta\right] g(0) + \int_\delta^1 dx\,\frac{g(x)}{x}
\end{aligned}$$

- We get the explicit pole in $\epsilon$ and a finite integral that can be computed numerically

# Subtraction method

$$\lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x) \qquad\qquad f(x)=\frac{g(x)}{x}$$

- Add and subtract the same term

$$\lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x) = \lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon}\left[\frac{g(0)}{x}+f(x)-\frac{g(0)}{x}\right]$$

$$= \lim_{\epsilon\to 0}\int_0^1 dx\left[g(0)\frac{x^{-\epsilon}}{x}+\frac{g(x)-g(0)}{x^{1+\epsilon}}\right]$$

$$= \lim_{\epsilon\to 0}\frac{-1}{\epsilon}g(0)+\int_0^1 dx\,\frac{g(x)-g(0)}{x}$$

- Like before, we have factored out the explicit divergence. The coefficient in front of the $1/\epsilon$ pole is the same in both methods (as it should be!)
- According to the KLN theorem the divergence cancels against the virtual corrections

# Slicing vs Subtraction

Slicing: $$\int_{\delta}^{1} dx \, \frac{g(x)}{x} + g(0) \log \delta$$

Subtraction: $$\int_{0}^{1} dx \, \frac{g(x) - g(0)}{x}$$

"Plus distribution"

- Terms of order $\delta$ are neglected in the slicing method; the subtraction method is exact
  - One has to proof that any observable is independent of $\delta$ when $\delta \to 0$
- Both methods feature cancellations between large numbers: if for an observable $O$, if $\lim_{x \to 0} O(x) \neq O(0)$ or we choose the bin-size too small, instabilities render the computation useless
  - We already knew that! KLN is sufficient; one must have infra-red safe observables and cannot ask for infinite resolution
- Subtraction method is more flexible -> method of choice in automation

# NLO with Subtraction

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m) + \int d^4\Phi_m \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_{m+1} \, R(\Phi_{m+1})$$

- With the subtraction method this is replace by

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m)$$

$$+ \int d^4\Phi_m \left[ \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon \to 0}$$

$$+ \int d^4\Phi_{m+1} \left[ R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

- Terms between the brackets are finite. Can integrate them numerically and independent from one another in 4 dimensions

# Subtraction methods

- $G(\overline{\Phi}_{m+1})$ should be defined such that

  1) it exactly matches the singular behavior of $R(\Phi_{m+1})$

  2) its form is convenient for MC integration techniques

  3) it is exactly integrable in d dimensions over the one-particle subspace $\int d^d\Phi_1 G(\overline{\Phi}_{m+1})$, leading to soft and/or collinear divergences as explicit poles in the dimensional regulator

  4) it is universal, i.e. "process independent"
  → "overall factor" times the Born process

# Two methods

- Catani-Seymour dipole subtraction
  - ☑ Most used method
  - ☑ Clear written paper on how to use this method in practice
  - ☑ Method evolved from cancellation of the soft divergence
  - ☑ Proven to work for simple as well as complicated processes
  - ☑ Automation in publicly available packages: MadDipole, AutoDipole, Helac-Dipoles, Sherpa
- FKS subtraction
  - ☑ Not so well-known
  - ☑ (Probably) more efficient, because less subtraction terms are needed
  - ☑ Collinear divergences as a starting point
  - ☑ Proven to work for simple as well as complicated processes
  - ☑ Preferred method when interfacing NLO to a parton shower
  - ☑ Implemented in MadFKS and (a) MC@NLO

# FKS subtraction

- FKS subtraction: Frixione, Kunszt & Signer 1996
- Also known as "residue subtraction"
- Based on using plus-distributions to regulate the infrared divergences of the real emission matrix elements
- Implemented in the MadGraph/MadEvent framework: MadFKS

# Phase-space partitions

- Easiest to understand by starting from real emission:

$$d\sigma^R = |M^{n+1}|^2 d\phi_{n+1}$$

- $|M^{n+1}|^2$ diverges like $\frac{1}{\xi_i^2}\frac{1}{1-y_{ij}}$ with $\xi_i = E_i/\sqrt{\hat{s}}$ $\quad y_{ij} = \cos\theta_{ij}$

- Partition the phase space in such a way that each partition has at most one soft and one collinear singularity

$$d\sigma^R = \sum_{ij} S_{ij} |M^{n+1}|^2 d\phi_{n+1} \qquad \sum_{ij} S_{ij} = 1$$

- Use plus distributions to regulate the singularities

$$d\tilde{\sigma}^R = \sum_{ij} \left(\frac{1}{\xi_i}\right)_+ \left(\frac{1}{1-y_{ij}}\right)_+ \xi_i(1-y_{ij}) S_{ij} |M^{n+1}|^2 d\phi_{n+1}$$

# Regularized by plus-prescription

$$d\tilde{\sigma}^R = \sum_{ij} \left(\frac{1}{\xi_i}\right)_+ \left(\frac{1}{1-y_{ij}}\right)_+ \xi_i(1-y_{ij}) S_{ij} |M^{n+1}|^2 d\phi_{n+1}$$

- Definition plus distribution

$$\int d\xi \left(\frac{1}{\xi}\right)_+ g(\xi) = \int d\xi \, \frac{g(\xi) - g(0)}{\xi}$$

- One event has maximally three counter events:
  - Soft: $\xi_i \to 0$
  - Collinear: $y_{ij} \to 1$
  - Soft-collinear: $\xi_i \to 0 \qquad y_{ij} \to 1$

# Regularized by plus-prescription

$$d\tilde{\sigma}^R = \sum_{ij} \left(\frac{1}{\xi_i}\right)_{\xi_{cut}} \left(\frac{1}{1-y_{ij}}\right)_{\delta_O} \xi_i(1-y_{ij}) S_{ij} |M^{n+1}|^2 d\phi_{n+1}$$

- Modified definition plus distribution (include counter terms only when event is close to being singular)

$$\int d\xi \left(\frac{1}{\xi}\right)_{\xi_{cut}} g(\xi) = \int d\xi \, \frac{g(\xi) - g(0)\Theta(\xi_{cut} - \xi)}{\xi}$$

- One event has maximally three counter events:
  - Soft: $\xi_i \to 0$
  - Collinear: $y_{ij} \to 1$
  - Soft-collinear: $\xi_i \to 0 \qquad y_{ij} \to 1$

# Subtraction terms

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[ \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon \to 0}$$
$$+ \int d^4\Phi_{m+1} \left[ R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

- This defines the subtraction terms for the reals
- They need to be integrated over the one-parton phase space (analytically) to get the explicit poles $1/\epsilon$ and added to the virtual corrections so that these poles cancel
  - these are process-independent terms proportional to the (color-linked) Borns
- All formulae can be found in the MadFKS paper, arXiv:0908.4247

# MadFKS matrix elements

$$\sigma^{\rm NLO} \sim \int d^4\Phi_m \, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[ \int_{\rm loop} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon\to 0}$$
$$+ \int d^4\Phi_{m+1} \left[ R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

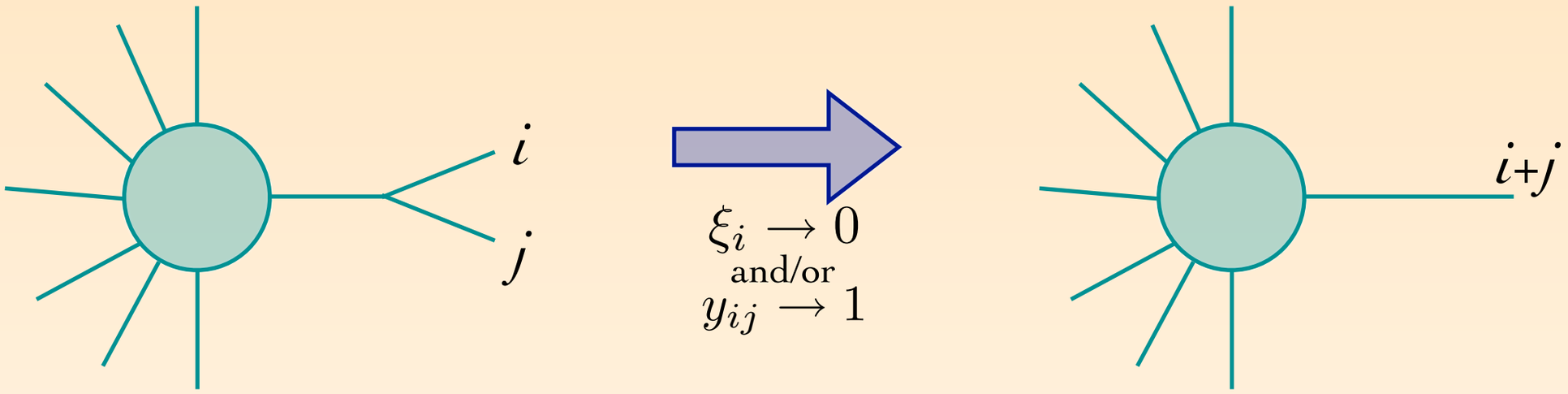
- When give the real-emission process to MadFKS
  - It will generates the matrix elements for the Born, real-emission and FKS subtraction terms
- Virtual corrections need to be provided by the user; linked by a library to the MadFKS code. When running, MadFKS tells the library for which phase-space points it should compute the virtual corrections

# Phase-space integration in MadFKS

$$\sigma^{\rm NLO} \sim \int d^4\Phi_m \, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[ \int_{\rm loop} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon \to 0}$$
$$+ \int d^4\Phi_{m+1} \left[ R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

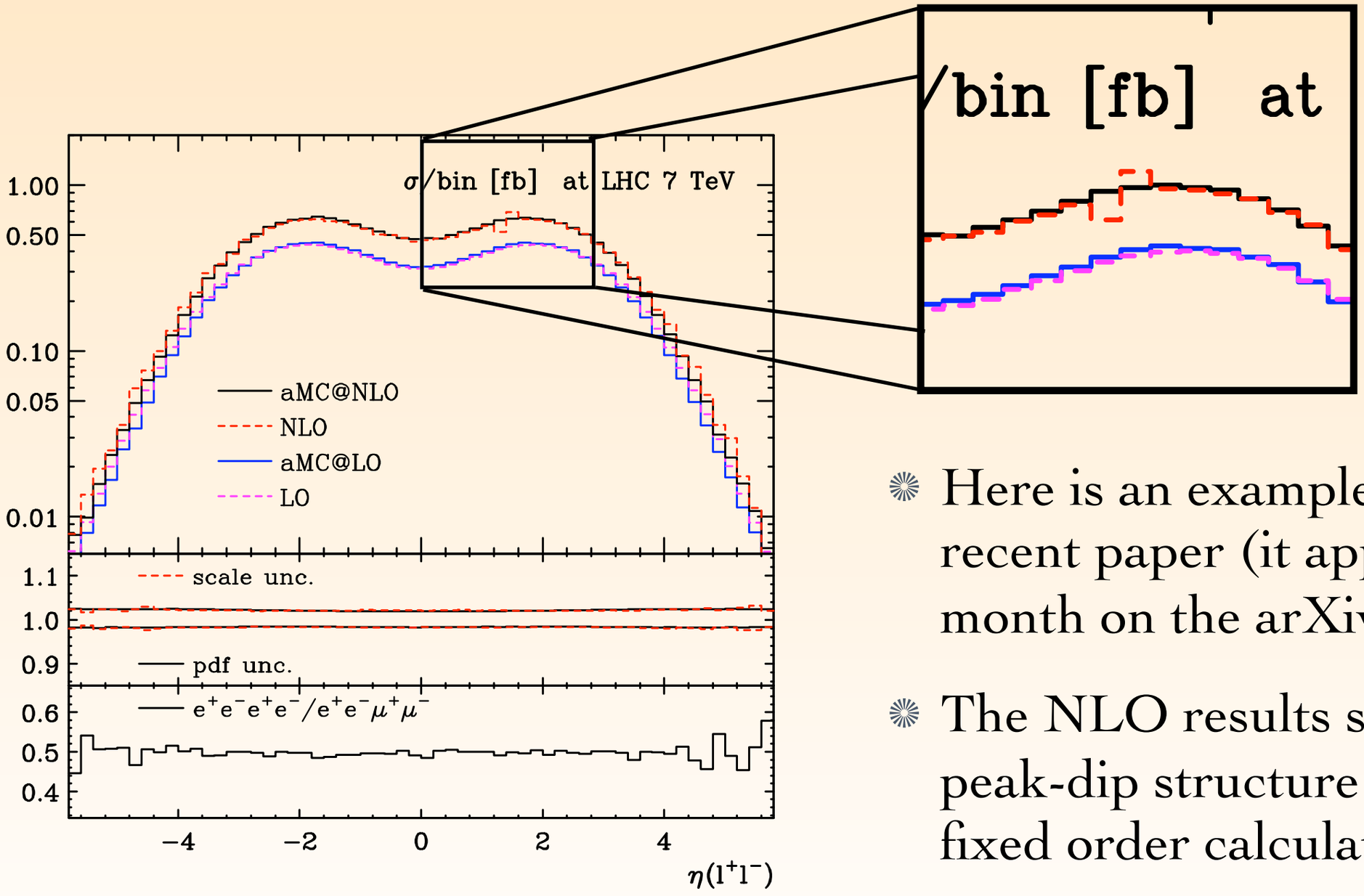
- Generate Born event using the standard MadGraph phase-space generation
- Generate the real-emission event, by choosing random numbers for the 3 extra integration variables: $d\xi_i$, $dy_{ij}$, $d\phi_i$
- Due to the phase-space partitioning, we know which singularities we need to subtract, generate the corresponding phase-space points, i.e. where $\xi_i=0$ or $y_{ij}=1$, or both
- Use these momenta to compute the terms in the NLO computations and perform the Monte Carlo integration

# Kinematics of counter events




- If $i$ and $j$ are two on-shell particles that are present in a splitting that leads to an singularity, for the counter events we need to combine their momenta to a new on-shell parton that's the sum of $i$+$j$
- This is not possible without changing any of the other momenta in the process
- When applying cuts or making plots, events and counter events might end-up in different bins
  - Use IR-safe observables and don't ask for infinite resolution! (KLN theorem)

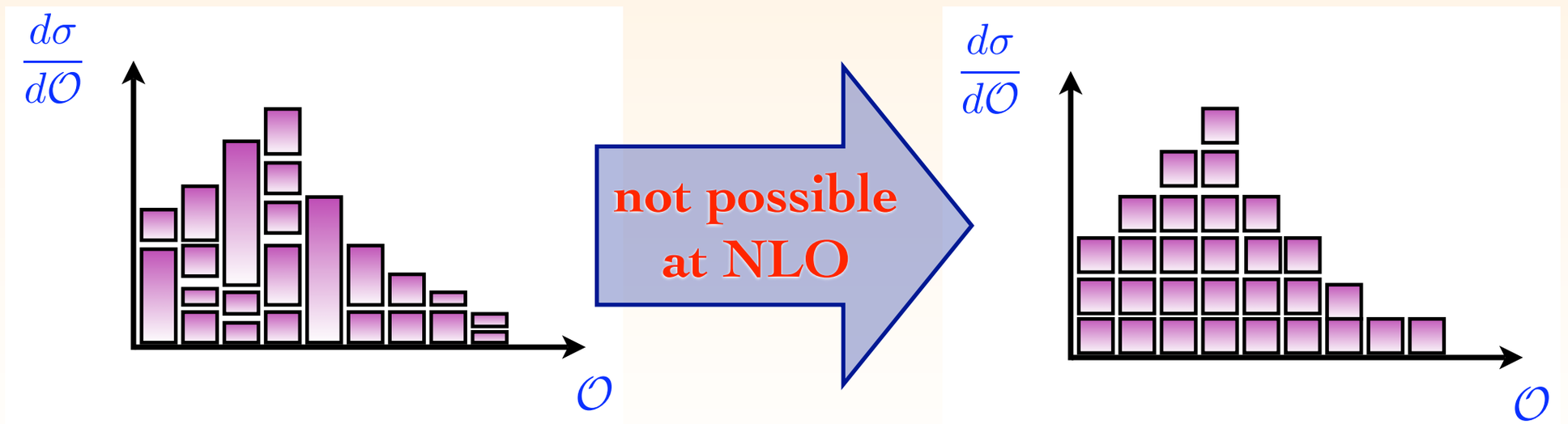
# Example in 4 charged lepton production




- Here is an example of a very recent paper (it appeared last month on the arXiv)
- The NLO results shows a typical peak-dip structure that hampers fixed order calculations

# Event unweighting?

- It is not possible to generate unweighted events in this set-up
  - Even though the integrals are finite, they are not bounded (compare with $\int_0^1 dx\, \frac{1}{\sqrt{x}}$ ), so there is no maximum to unweight against: a single event can have an arbitrarily large weight
  - Furthermore, event and counter event have different kinematics: which one to use for the unweighted event?



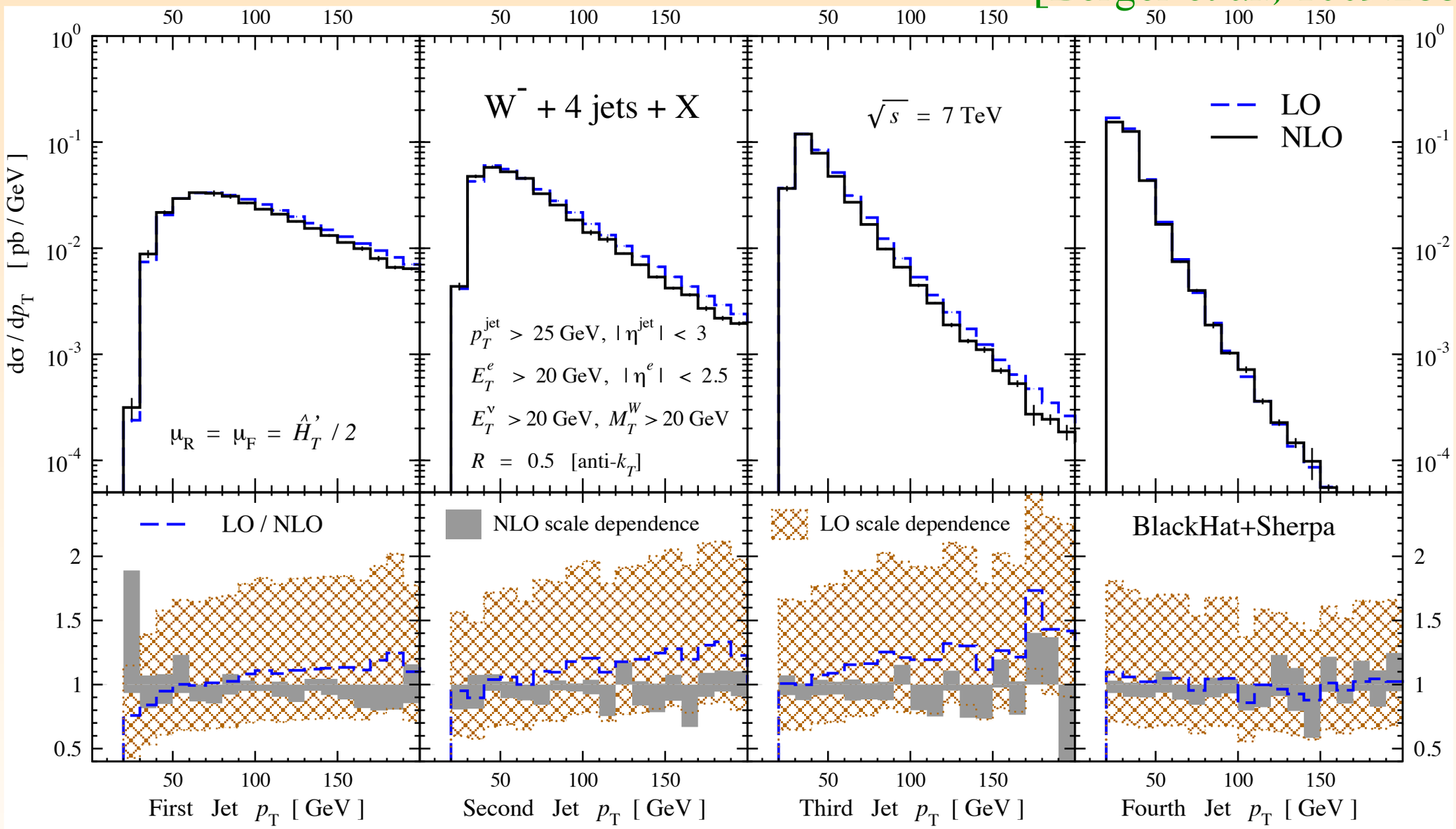
# Filling histograms on-the-fly

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[ \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon \to 0}$$
$$+ \int d^4\Phi_{m+1} \left[ R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

- In practice, when we do the MC integration we generate 2 sets of momenta
  1. An *m*-body set (for the Born, virtual and integrated counter terms)
  2. An *m*+1-body (for the NLO) which we map to the counter term momenta (for the counter terms)
- We compute the above formula; and apply cuts and fill histograms using the momenta corresponding to each term with the weight of that corresponding term

# W+4J AT NLO

[Berger et al., 1009.2338]



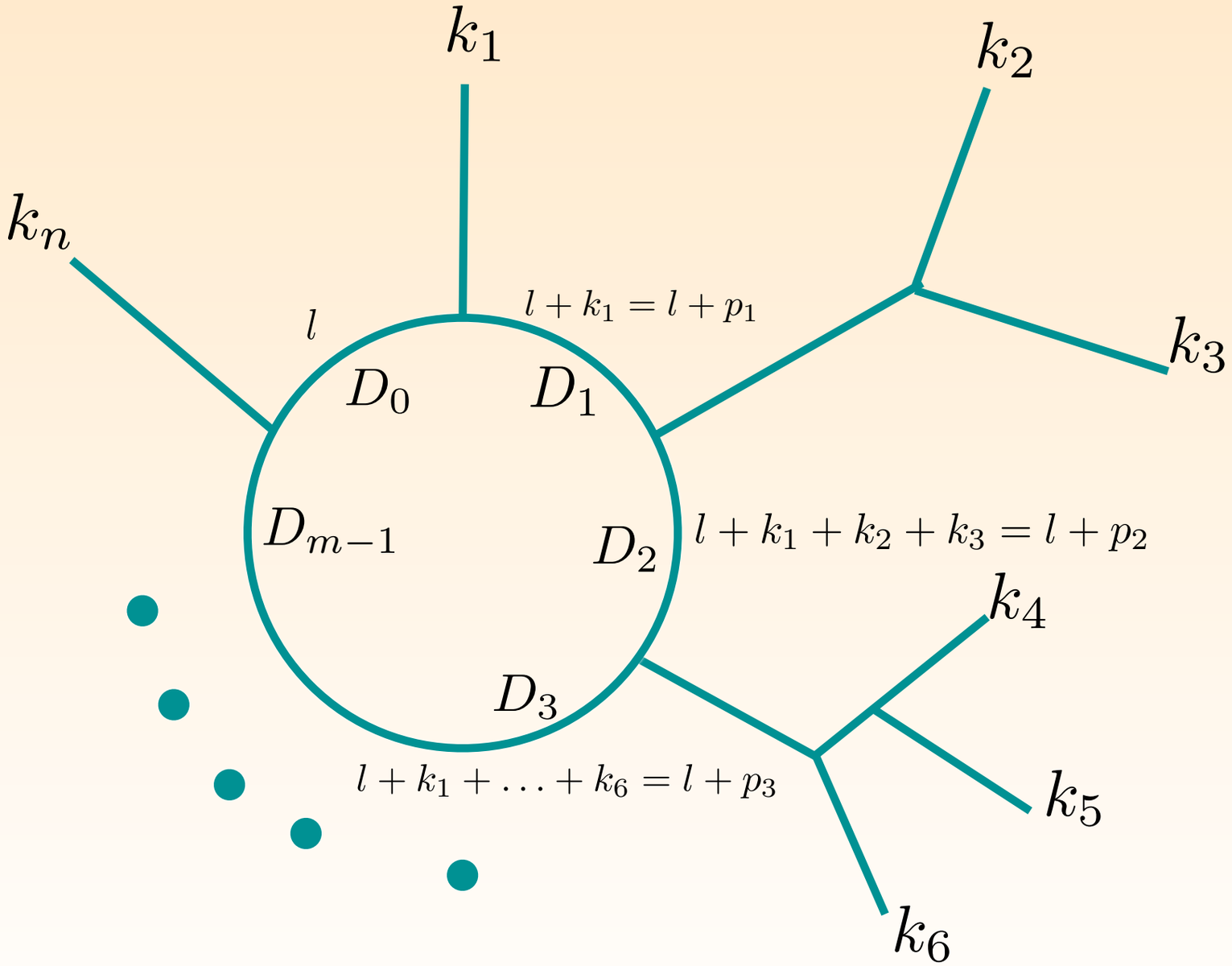

- W+4j production at the LHC: world record NLO computation
- Uncertainty (due to scale dependence) is greatly reduced at NLO

# Summary

- Both the virtual and real-emission corrections are IR divergent, but their sum is finite
- We can use the slicing or subtraction methods to factor the divergences in the real-emission phase-space integration and cancel them explicitly against the terms in the virtual corrections
  - Preferred method is the subtraction method (no approximations needed and proven to work very well for complicated processes)
- This generates events and counter events with slightly different kinematics
- When making plots or applying cuts, use only IR safe observables with finite resolution
- Phase-space integrals are finite, but not bounded: cannot unweight the events

# Loops

# ONE-LOOP INTEGRAL




- Consider this $m$-point loop diagram with $n$ external momenta
- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

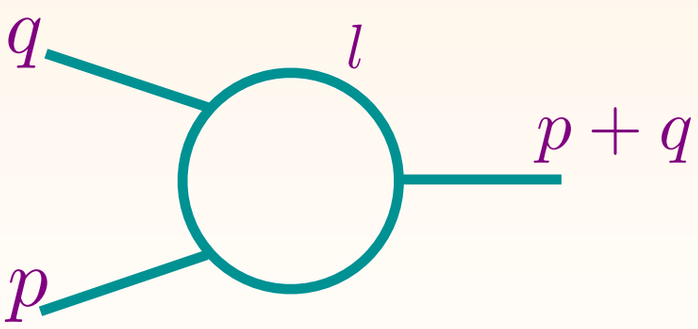
# Standard Approach

- Passarino-Veltman reduction:

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}} \rightarrow \sum_i \text{coeff}_i \int d^d l \, \frac{1}{D_0 D_1 \cdots}$$

  - Reduce a general integral to "scalar integrals" by "completing the square"

- Let's do an example:
  Suppose we want to calculate this triangle integral




$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$


- The only independent four vectors are $p^\mu$ and $q^\mu$ . Therefore, the integral must be proportional to those. We can set-up a system of linear equations and try to solve for $C_1$ and $C_2$

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

- We can solve for $C_1$ and $C_2$ by contracting with $p$ and $q$

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

where $[2l \cdot p] = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot p}{l^2 (l+p)^2 (l+q)^2}$ (For simplicity, the masses are neglected here)

- By expressing $2l.p$ and $2l.q$ as a sum of denominators we can express $R_1$ and $R_2$ as a sum of simpler integrals, e.g.

$$R_1 = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot p}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^n l}{(2\pi)^n} \frac{(l+p)^2 - l^2 - p^2}{l^2 (l+p)^2 (l+q)^2}$$

$$= \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+q)^2} - \int \frac{d^n l}{(2\pi)^n} \frac{1}{(l+p)^2 (l+q)^2} - p^2 \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- And similarly for $R_2$

$$R_2 = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot q}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^n l}{(2\pi)^n} \frac{(l+q)^2 - l^2 - q^2}{l^2 (l+p)^2 (l+q)^2}$$
$$= \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2} - \int \frac{d^n l}{(2\pi)^n} \frac{1}{(l+p)^2 (l+q)^2} - q^2 \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- Now we can solve the equation

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

by inverting the "Gram" matrix $G$

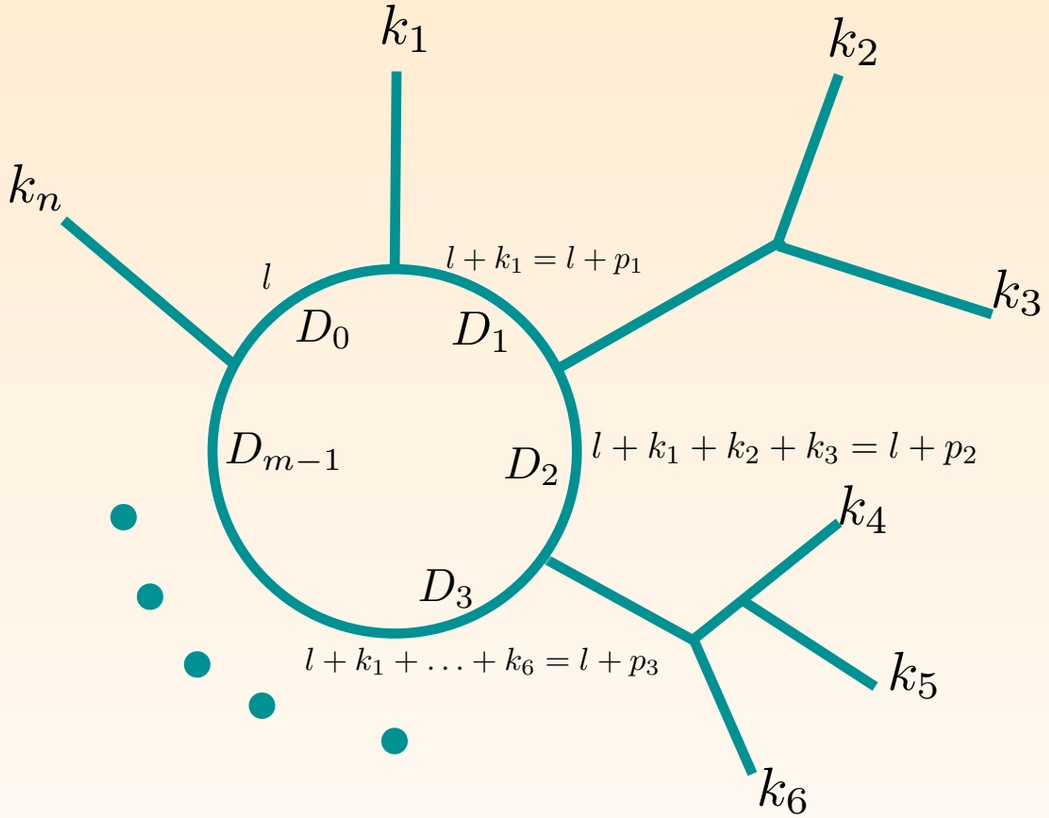
$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = G^{-1} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

and we have expressed our original integral

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

in terms of known, simpler integrals and we are done!

# Higher point integrals




- For loop integrals with many legs, the reduction to scalar integrals can still be performed
- Only up to 4-point scalar integrals are needed (in 4 dimensions)!
- The proof is beyond the scope of these lectures (it is straight forward by using the Van Neerven-Vermaseren basis for the loop momentum); it is related to the fact that in 4 dimensions only four 4-vectors can be linearly independent

# Basis of scalar integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3}\,\text{Box}_{i_0 i_1 i_2 i_3}$$
$$+ \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2}\,\text{Triangle}_{i_0 i_1 i_2}$$
$$+ \sum_{i_0<i_1} b_{i_0 i_1}\,\text{Bubble}_{i_0 i_1}$$
$$+ \sum_{i_0} a_{i_0}\,\text{Tadpole}_{i_0}$$
$$+ R + \mathcal{O}(\epsilon)$$

- The a, b, c, d and R coefficients depend only on external parameters and momenta

$$D_i = (l+p_i)^2 - m_i^2$$

$$\text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$
$$\text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$
$$\text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$
$$\text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

- All these scalar integrals are known and available in computer libraries (FF [v. Oldenborgh], QCDLoop [Ellis, Zanderighi], OneLOop [v. Hameren])

# Divergences

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

$$D_i = (l + p_i)^2 - m_i^2$$

$$\text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$

$$\text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$

$$\text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$

$$\text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

- The coefficients d, c, b and a are finite and do not contain poles in $1/\epsilon$
- The $1/\epsilon$ dependence is in the scalar integrals (and the UV renormalization)
- When we have solved this system (and included the UV renormalization) we have the full dependence on the soft/collinear divergences in terms of coefficients in front of the poles. These divergences should cancel against divergences in the real emission corrections (according to KLN theorem)

$$\text{Virtual} \sim v_0 + \frac{v_1}{\epsilon} + \frac{v_2}{\epsilon^2}$$

# About the R term

- In our example the decomposition to scalar integrals was "exact", i.e. there were no left-over terms.
- This is true for most integrals. Only if the rank of the integral is

$$r \geq \max\Big\{(N-1), 2)\Big\}$$

there are some extra contributions which are called "Rational terms" that are not proportional to a scalar integral

- They are of UV origin and come from the $\epsilon$ (dimensional regulator) dependence of the integral times a scalar integral that is UV divergent

$$\text{Rational terms } \sim \epsilon B_0(p, m_1, m_2)$$

(The Bubble integrals are the only UV divergent integrals)

- When taking the limit $\epsilon \to 0$, only the leading contribution remains, which are independent from the scalar integral itself

# Automation

- Advantage:
  - The method above can be straight-forwardly generalized to any one-loop integral (appearing in a renormalizable theory)
- Disadvantage:
  - For relatively simple processes, the number of terms already explodes (several 100 MB of code is no exception for the matrix elements of a $2 \to 3$ process); simplifications require hard work and are difficult to do in a general way
  - Does only work when the integrals are known analytically

# The "NLO revolution"

## One indicator of NLO progress

| | | |
|---|---|---|
| pp → W + 0 jet | 1978 | Altarelli, Ellis, Martinelli |
| pp → W + 1 jet | 1989 | Arnold, Ellis, Reno |
| pp → W + 2 jets | 2002 | Campbell, Ellis |
| pp → W + 3 jets | 2009 | BH+Sherpa |
| | | Ellis, Melnikov, Zanderighi |
| pp → W + 4 jets | 2010 | BH+Sherpa |

Slide from Lance Dixon

# New loop techniques

- The "loop revolution": new techniques for computing one-loop matrix elements are now established:
  - Generalized unitarity (e.g. BlackHat, Rocket, ...)
    [Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008;...]
  - Integrand reduction (e.g. CutTools, GoSam)
    [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - Tensor reduction (e.g. Golem)
    [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008]

# Integrand reduction

- Any one-loop integral can be decomposed in scalar integrals
- The task is to find these coefficients efficiently (analytically or numerically)
- The integrand (or OPP [Ossola, Papadopoulos, Pittau 2006]) reduction method is a method that has been automated in the **CutTools** program to find these coefficients in an automated way
- The integrand reduction technique is what we have adopted to use in MadGraph to compute the loop diagrams

# At the integrand level

- The decomposition to scalar integrals presented before works at the level of the integrals
- If we would know a similar relation at the **integrand** level, we would be able to manipulate the integrands and extract the coefficients without doing the integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

- This is exactly what the OPP reduction does
  - The decomposition is the same, except that there might be contributions that integrate to zero

# At the integrand level

- Consider, e.g., the Box coefficient:

$$d_{i_0 i_1 i_2 i_3}\mathrm{Box}_{i_0 i_1 i_2 i_3} = d_{i_0 i_1 i_2 i_3}\int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

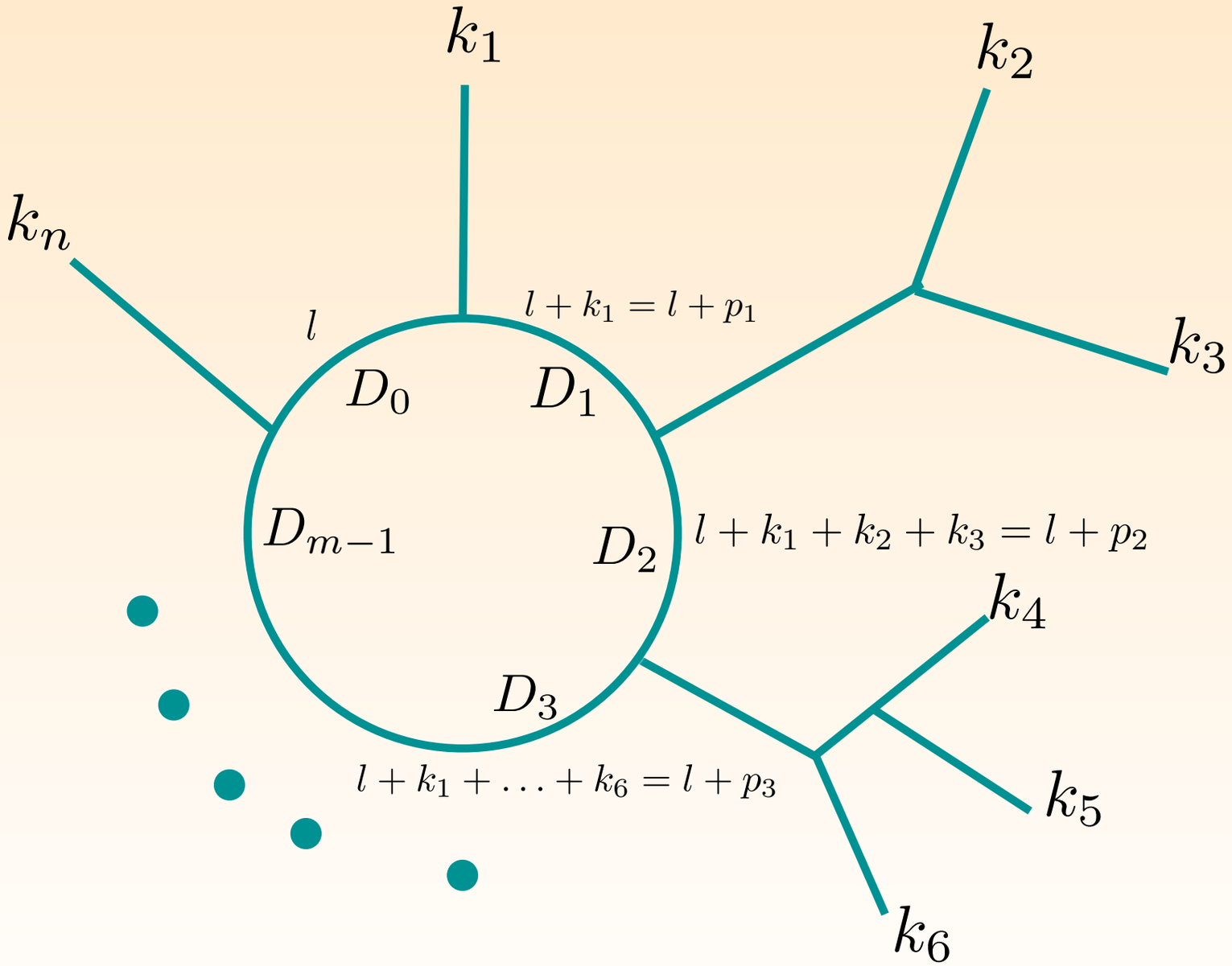
$$= \int d^d l \frac{d_{i_0 i_1 i_2 i_3}}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

$$= \int d^d l \frac{d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

$$\text{where} \quad \int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}} = 0$$

- And similarly for the $c$, $b$, $a$ and $R$ terms
- The contributions that vanish when doing the integral are called "spurious terms"

# ONE-LOOP INTEGRAL




- Consider this $m$-point loop diagram with $n$ external momenta
- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

# OPP DECOMPOSITION

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

- For the numerator of any integrand of a one-loop computation we can therefore write

$$\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[ d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[ c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[ b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[ a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}$$

# Numerical evaluation

- By choosing specific values for the loop momentum $l$, we end up with a system of linear equations
  - In a renormalizable theory, the rank of the integrand is always smaller (or equal) to the number of particles in the loop (with a conveniently chosen gauge)
  - We can straight-forwardly set the it up by sampling the numerator numerically for various values of the loop momentum $l$
  - By choosing $l$ smartly, the system greatly reduces
    - In particular when we chose $l$ to be a complex 4-vector

# Functional form of the spurious terms

- The functional form of the spurious terms is known (it depends on the rank of the integral and the number propagators in the loop) [del Aguila, Pittau 2004]
  - for example, a box coefficient from a rank 1 numerator is

$$\tilde{d}_{i_0 i_1 i_2 i_3}(l) = \tilde{d}_{i_0 i_1 i_2 i_3}\, \epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma$$

(remember that $\rho_i$ is the sum of the momentum that has entered the loop so far, so we always have $\rho_0 = 0$)

- The integral is zero

$$\int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_0 D_1 D_2 D_3} = \tilde{d}_{i_0 i_1 i_2 i_3} \int d^d l \frac{\epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma}{D_0 D_1 D_2 D_3} = 0$$

# How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing $l$ such that

$$D_0(l^{\pm}) = D_1(l^{\pm}) = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$

sets all the terms in this equation to zero except the first line

There are two (complex) solutions to this equation due to the quadratic nature of the propagators

# How it works...

$$N(l^{\pm}) = d_{0123} + \tilde{d}_{0123}(l^{\pm}) \prod_{i \neq 0,1,2,3}^{m-1} D_i(l^{\pm})$$

- Two values are enough given the functional form for the spurious term. We can immediately determine the Box coefficient

$$d_{0123} = \frac{1}{2}\left[\frac{N(l^{+})}{\prod_{i\neq 0,1,2,3}^{m-1} D_i(l^{+})} + \frac{N(l^{-})}{\prod_{i\neq 0,1,2,3}^{m-1} D_i(l^{-})}\right]$$

- By choosing other values for $l$, that set other combinations of 4 "denominators" to zero, we can get all the Box coefficients

# How it works...

- Now that we have all the Box coefficients we can start choosing values for *l* that set 3 "denominators" to zero to get the Triangle coefficients. Of course, now both the first and the second lines contribute.

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

- We already have solved the coefficients of the first line in the previous iteration, so also here there is only a simple system of equations to solve
- Once we have all the Triangle coefficients, we can continue to determine the Bubble coefficients; and finally the Tadpole coefficients

# How it works...

- For each phase-space point we have to solve the system of equations
- Due to the fact that the system reduces when picking special values for the loop momentum, the system greatly reduces
- We can decompose the system at the level of the amplitude, diagram or in between. As long as we provide the corresponding numerator function. In MadGraph we decompose diagram by diagram, but we are considering improvements
- For a given phase-space point, we have to compute the numerator function several times (~50 or so for a $2 \to 3$ process)

# A CLASSICAL EXAMPLE

- Suppose we want to compute this integral $$\int d^d l \frac{1}{D_0 D_1 D_2 D_3 D_4 D_5 D_6}$$
- So we that the numerator is $N(l) = 1$ $$D_i = (l + p_i)^2 - m_i^2$$
- We know that we need only Box, Triangle, Bubble (and Tadpole) contributions. Let's find the first Box integral coefficient.

- Take the two solutions of
$$D_0(l^{\pm}) = D_1(l^{\pm}) = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$
- And use the relation we found before and we directly have
$$d_{0123} = \frac{1}{2}\left[\frac{1}{D_4(l^+)D_5(l^+)D_6(l^+)} + \frac{1}{D_4(l^-)D_5(l^-)D_6(l^-)}\right]$$

$$\ldots$$

# Complications in d dimensions

- In the previous consideration I was very sloppy in considering if we are working in 4 or d dimensions
- In general, external momenta and polarization vectors are in 4 dimensions; only the loop momentum is in d dimensions
- To be more correct, we compute the integral

$$\int d^d l \, \frac{N(l,\tilde{l})}{\bar{D}_0 \bar{D}_1 \bar{D}_2 \cdots \bar{D}_{m-1}}$$

$$\bar{l} = l + \tilde{l}$$

d dim  4 dim  epsilon dim

$$\bar{D}_i = (\bar{l} + p_i)^2 - m_i^2 = (l + p_i)^2 - m_i^2 + \tilde{l}^2 = D_i + \tilde{l}^2$$

$$l \cdot \tilde{l} = 0 \qquad \bar{l} \cdot p_i = l \cdot p_i \qquad \bar{l} \cdot \bar{l} = l \cdot l + \tilde{l} \cdot \tilde{l}$$

# Implications

$$\sum_{0\le i_0<i_1<i_2<i_3}^{m-1} d(i_0i_1i_2i_3)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}\bar{D}_{i_3}}$$

$$+\sum_{0\le i_0<i_1<i_2}^{m-1} c(i_0i_1i_2)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}}$$

$$+\sum_{0\le i_0<i_1}^{m-1} b(i_0i_1)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}}$$

$$+\sum_{i_0=0}^{m-1} a(i_0)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}}$$

$$+\,R\,.$$

- The decomposition in terms of scalar integrals has to be done in d dimensions
- This is why the rational part $R$ is needed

# Rational terms

- The main difference is how we get the rational terms (we already saw them in the Passarino-Veltman reduction)
- In the OPP method, they are split into two contributions, generally called

$$R = R_1 + R_2$$

- Both have their origin in the UV part of the model, but only $R_1$ can be directly computed in the OPP reduction and is given by the CutTools program

# $R_1$

- The origin of $R_1$ is coming is the denominators of the propagators in the loop

$$\frac{1}{D_i} \to \frac{1}{\bar{D}_i} = \frac{1}{D}\left(1 - \frac{\tilde{l}^2}{D_i}\right)$$

- Of course, the propagator structure is known, so these contributions can be included in the OPP reduction

- They give contributions proportional to

$$\int d^d\bar{l} \frac{\tilde{l}^2}{\bar{D}_i \bar{D}_j} = -\frac{i\pi^2}{2}\left[m_i^2 + m_j^2 - \frac{(p_i - p_j)^2}{3}\right] + \mathcal{O}(\epsilon)$$

$$\int d^d\bar{l} \frac{\tilde{l}^2}{\bar{D}_i \bar{D}_j \bar{D}_k} = -\frac{i\pi^2}{2} + \mathcal{O}(\epsilon)$$

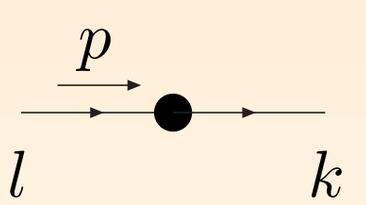
$$\int d^d\bar{l} \frac{\tilde{l}^4}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_l} = -\frac{i\pi^2}{6} + \mathcal{O}(\epsilon)$$

# $R_2$

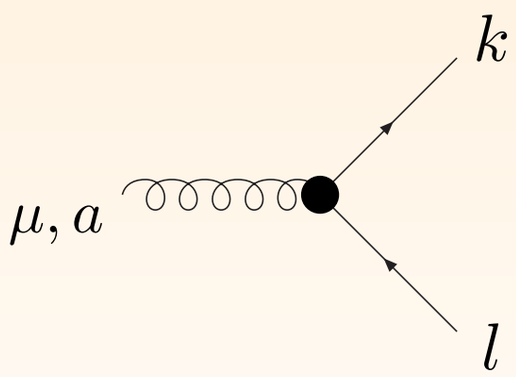
- The other origin of rational terms is the numerator itself. For integrals with rank > 2 we can have dependence in the numerator that is proportional to $\bar{l}^2$
- Unfortunately, this dependence can be quite hidden; maybe it is only explicitly there after doing the Clifford algebra
- Because we want to solve the system without doing this algebra analytically (we want to solve it numerically) we cannot get these contributions directly within the OPP reduction
- Within a given model, there is only a finite number of sources that can give these contributions; They have all been identified within the SM, and can be computed with the "$R_2$ counter terms"

# $R_2$ Feynman rules

- Given that the $R_2$ contributions are of UV origin, only up to 4-point functions contribute to it (in a renormalizable theory)
- They can be computed using special Feynman rules, similarly to the UV counter term Feynman rules needed for the UV renormalization, e.g.




$$= \frac{ig^2}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\delta_{kl}(-\not{p}+2m_q)\,\lambda_{HV}$$




$$= \frac{ig^3}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\,t^a_{kl}\gamma_\mu\,(1+\lambda_{HV})$$

[Draggiotis, Garzelli, Papadopoulos, Pittau]

- Unfortunately these Feynman rules are model dependent.
  ⇒ Maybe we can use FeynRules+FeynArts to compute them for any model?

# Improvement over Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.
  - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum $l$, depending on the kinematics of the event ☺
  - Choosing $l$ such that internal propagators go on-shell, enormously simplifies the resulting system ☺
  - OPP reduction is implemented in CutTools (publicly available). Given the integrand, CutTools provides all the coefficients in front of the scalar integrals and the $R_1$ term ☺
  - Analytic information is needed for the $R_2$ term, but can be compute once and for all for a given model ☹☺

# In MadGraph

- MadGraph is very good at giving numerical expressions for matrix elements. Exactly what is needed by CutTools to get the coefficients of the scalar integrals
- However, it is only tree-level...
- Needed to create an add-on to MadGraph to generate loop diagrams: MadLoop!

# MadLoop

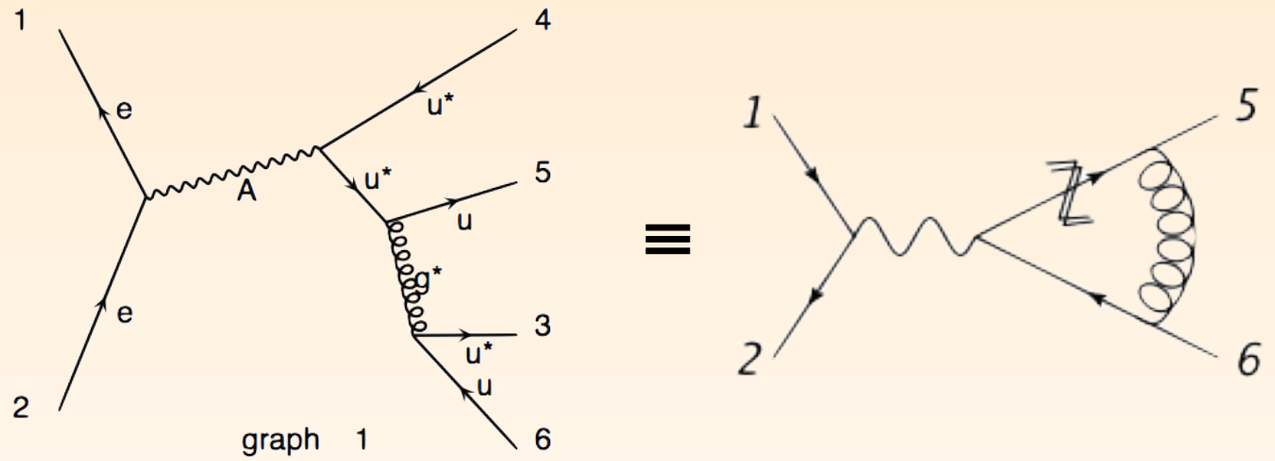
- Instead of writing a new code to generate loop diagrams, we use the existing, well-tested MadGraph code to generate tree-level diagrams

- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow u^* \bar{u}^* u \bar{u}$ (loop particles are denoted with a star). MadGraph will generate 8 L-cut diagrams. Here are two of them:

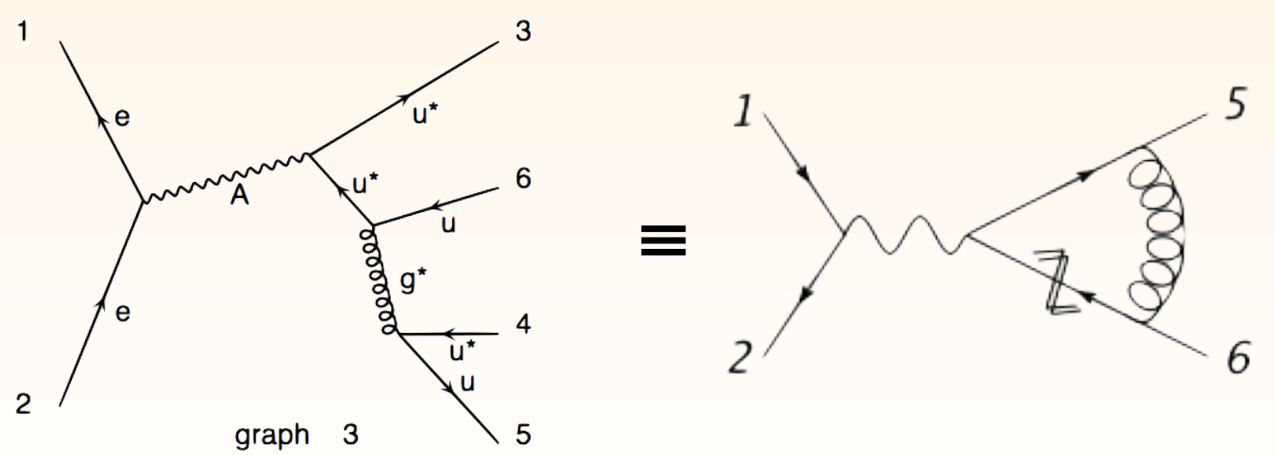
  - All diagrams with two extra particles are generated and the ones that are needed are filtered out

  - Each diagram gets an unique tag: any mirror and/or cyclic permutations of tags of diagrams already in the set are taken out

  - Additional filter to eliminate tadpoles and bubbles attached to external lines




$$\text{Diag}_1 = [u^*(6)g^*(5)u^*(A)]$$




$$\text{Diag}_3 = [u^*(A)u^*(6)g^*(5)]$$

# MadLoop

- Several new features needed to be implemented in MadGraph
  - Recognition of the loop topologies in order to filter L-cut diagrams
  - Structure to deal with two MadGraph processes simultaneously (L-cut and Born-like)
  - Treat the color to obtain the correct interference between the Born and the loop diagrams
  - Special form of the integrand for CutTools: no propagator denominators, complex momenta and reconstruction of the missing propagator for sewed particles (e.g., when L-cut particle is a gluon, $\sum \epsilon^{\mu}(p)\epsilon^{\nu}(p) \rightarrow g^{\mu\nu}$ )
  - Implementation of QCD ghosts
  - Implementation of the special vertices for the rational part $R_1$ and the UV renormalization

# MadLoop: Exceptional PS points

- There are (almost) always phase-space points for which the numerical reduction to determine the coefficients in front of the scalar integrals does not work due to numerical instabilities
- CutTools has build-in routines to determine if a phase-space point is exceptional or not
  - CT can ask MadLoop to evalutate the integrand at a given loop momentum and check if the result is close enough to the one from the reconstructed integrand
  - By sending $m_i^2 \to m_i^2 + M^2$ CT has an independent reconstruction of the numerator and can check if both match
- Using quadruple precision numerics in the reduction helps, but not always

# MadLoop: Exceptional PS points

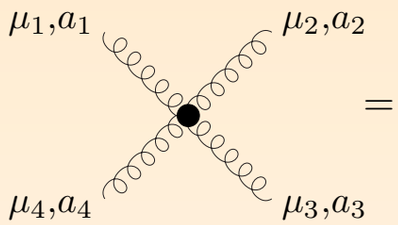
- When CutTools assigns a phase-space point to be unstable, MadLoop tries to cure it
  - Check if the Ward Identity holds at a satisfactory level
  - Shift the phase-space point by rescaling one of the components of the 3-momenta (for all particles), e.g. $k_i^3 = (1 + \lambda_\pm) k_i^3$ , and adjusting the energy components to keep the point on-shell
    - Provide an estimate of the virtual of the original phase-space point (with uncertainty) $V_{\lambda=0}^{FIN} = \left|\mathcal{A}_{\lambda=0}^{born}\right|^2 (c \pm \Delta)$ where

$$c = \frac{1}{2}\left(v_{\lambda_+}^{FIN} + v_{\lambda_-}^{FIN}\right) \qquad \Delta = \left|v_{\lambda_+}^{FIN} - v_{\lambda_-}^{FIN}\right| \qquad v_{\lambda_\pm}^{FIN} = \frac{V_{\lambda_\pm}^{FIN}}{|\mathcal{A}_{\lambda=0}^{born}|^2}$$

  - If all shifts fail (very rarely) use the median of the results of the last 100 stable points and the median absolute deviation to determine the associated uncertainty

# MadLoop: limitations

- Of course, there are some limitations on what the code cannot do yet...
  - No four-gluon vertex at the Born level: the special vertex to compute the remainder is too complicated to implement in MadGraph v4




$$= -\frac{ig^4 N_{col}}{96\pi^2} \sum_{P(234)} \left\{ \left[ \frac{\delta_{a_1 a_2}\delta_{a_3 a_4} + \delta_{a_1 a_3}\delta_{a_4 a_2} + \delta_{a_1 a_4}\delta_{a_2 a_3}}{N_{col}} \right.\right.$$

$$+ 4\,Tr(t^{a_1}t^{a_3}t^{a_2}t^{a_4} + t^{a_1}t^{a_4}t^{a_2}t^{a_3})\,(3+\lambda_{HV})$$

$$\left. - Tr(\{t^{a_1}t^{a_2}\}\{t^{a_3}t^{a_4}\})\,(5+2\lambda_{HV}) \right] g_{\mu_1\mu_2} g_{\mu_3\mu_4}$$

$$\left. + 12 \frac{N_f}{N_{col}} Tr(t^{a_1}t^{a_2}t^{a_3}t^{a_4}) \left( \frac{5}{3} g_{\mu_1\mu_3} g_{\mu_2\mu_4} - g_{\mu_1\mu_2} g_{\mu_3\mu_4} - g_{\mu_2\mu_3} g_{\mu_1\mu_4} \right) \right\}$$

  - If EW bosons appear in the loops, the reduction by CutTools might not work because we use gauge a physical gauge (rank of diagrams can become too large)
  - No finite-width effects for massive particles also appearing in the loops
  - All Born contributions must factorize the same power of all coupling orders

# On-going improvements

- The MadLoop code is being rewritten in MadGraph v5. This will:
  - remove the limitations presented on the previous slide
  - make it faster:
    - Recycling of tree-structures attached to the loops
      - Identify identical contributions (e.g. massless fermion loops of different flavors)
      - Call CutTools not per diagram, but per set of diagrams with the same loop kinematics
      - Use recursion relations (will mostly help the real-emission corrections)
  - allow for the automatic generation of UV renormalization and remainder vertices using FeynRules [*Christensen, Duhr et al.*] for BSM physics

# Local checks

| $u\bar{u} \to W^+W^-b\bar{b}$ | MadLoop | Ref. [33] |
|---|---|---|
| $a_0$ | 2.338047209268890E-008 | 2.338047130649064E-008 |
| $c_{-2}$ | -2.493920703542680E-007 | -2.493916939359002E-007 |
| $c_{-1}$ | -4.885901939046758E-007 | -4.885901774740355E-007 |
| $c_0$ | -2.775800623041098E-007 | -2.775787767591390E-007 |
| $gg \to W^+W^-b\bar{b}$ | | |
| $a_0$ | 1.549795815702494E-008 | 1.549794572435312E-008 |
| $c_{-2}$ | -2.686312747217639E-007 | -2.686310592221201E-007 |
| $c_{-1}$ | -6.078687041491385E-007 | -6.078682316434646E-007 |
| $c_0$ | -5.519004042667462E-007 | -5.519004727276688E-007 |

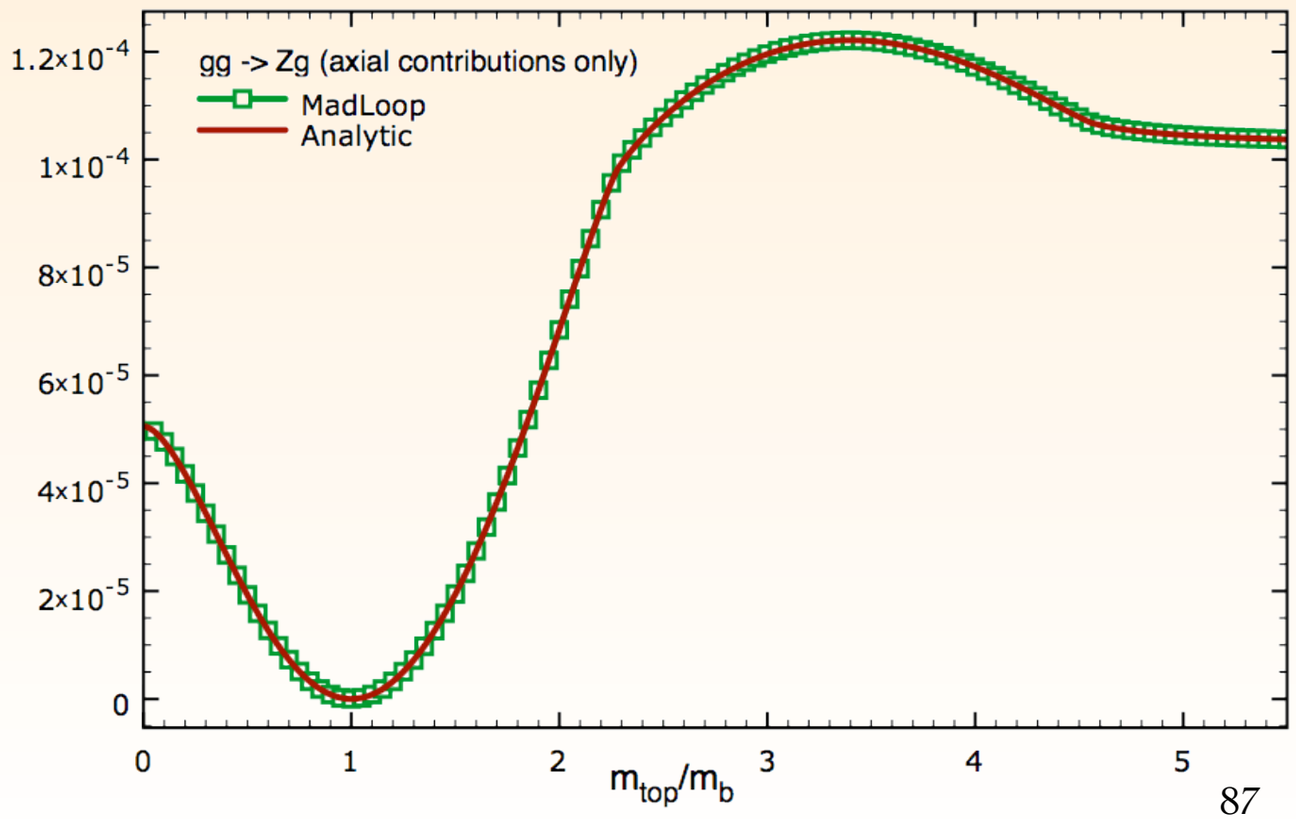
Ref. [33]: *A. van Hameren et al. arXiv:0903.4665*

The numerics are pin-point on analytical data, even with several mass scales.

Analytic computation via an implementation of the formulae found in a paper by *J.J. van der Bij & N. Glover*

~25 processes checked against known results (24 pages appendix of MadLoop paper, arXiv:1103.0621)

We believe the code is very robust - e.g., MadLoop helped to find mistakes in published NLO computations implementations ($pp \to Zjj$, $pp \to W^+W^+jj$)



# Integrated results

- Errors are the MC integration uncertainty only
- Cuts on jets, $\gamma^*/Z$ decay products and photons, but no cuts on b quarks (their mass regulates the IR singularities)
- Efficient handling of exceptional phase-space points: their uncertainty always at least two orders of magnitude smaller than the integration uncertainty
- Running time: two weeks on ~150 node cluster leading to rather small integration uncertainties
- MadFKS+MadLoop results are fully differential in the final states (but only parton-level)

| | Process | $\mu$ | $n_{lf}$ | Cross section (pb) LO | Cross section (pb) NLO |
|---|---|---|---|---|---|
| a.1 | $pp \to t\bar{t}$ | $m_{top}$ | 5 | $123.76 \pm 0.05$ | $162.08 \pm 0.12$ |
| a.2 | $pp \to tj$ | $m_{top}$ | 5 | $34.78 \pm 0.03$ | $41.03 \pm 0.07$ |
| a.3 | $pp \to tjj$ | $m_{top}$ | 5 | $11.851 \pm 0.006$ | $13.71 \pm 0.02$ |
| a.4 | $pp \to t\bar{b}j$ | $m_{top}/4$ | 4 | $25.62 \pm 0.01$ | $30.96 \pm 0.06$ |
| a.5 | $pp \to t\bar{b}jj$ | $m_{top}/4$ | 4 | $8.195 \pm 0.002$ | $8.91 \pm 0.01$ |
| b.1 | $pp \to (W^+ \to)e^+\nu_e$ | $m_W$ | 5 | $5072.5 \pm 2.9$ | $6146.2 \pm 9.8$ |
| b.2 | $pp \to (W^+ \to)e^+\nu_e\, j$ | $m_W$ | 5 | $828.4 \pm 0.8$ | $1065.3 \pm 1.8$ |
| b.3 | $pp \to (W^+ \to)e^+\nu_e\, jj$ | $m_W$ | 5 | $298.8 \pm 0.4$ | $300.3 \pm 0.6$ |
| b.4 | $pp \to (\gamma^*/Z \to)e^+e^-$ | $m_Z$ | 5 | $1007.0 \pm 0.1$ | $1170.0 \pm 2.4$ |
| b.5 | $pp \to (\gamma^*/Z \to)e^+e^-\, j$ | $m_Z$ | 5 | $156.11 \pm 0.03$ | $203.0 \pm 0.2$ |
| b.6 | $pp \to (\gamma^*/Z \to)e^+e^-\, jj$ | $m_Z$ | 5 | $54.24 \pm 0.02$ | $56.69 \pm 0.07$ |
| c.1 | $pp \to (W^+ \to)e^+\nu_e b\bar{b}$ | $m_W + 2m_b$ | 4 | $11.557 \pm 0.005$ | $22.95 \pm 0.07$ |
| c.2 | $pp \to (W^+ \to)e^+\nu_e t\bar{t}$ | $m_W + 2m_{top}$ | 5 | $0.009415 \pm 0.000003$ | $0.01159 \pm 0.00001$ |
| c.3 | $pp \to (\gamma^*/Z \to)e^+e^- b\bar{b}$ | $m_Z + 2m_b$ | 4 | $9.459 \pm 0.004$ | $15.31 \pm 0.03$ |
| c.4 | $pp \to (\gamma^*/Z \to)e^+e^- t\bar{t}$ | $m_Z + 2m_{top}$ | 5 | $0.0035131 \pm 0.0000004$ | $0.004876 \pm 0.000002$ |
| c.5 | $pp \to \gamma t\bar{t}$ | $2m_{top}$ | 5 | $0.2906 \pm 0.0001$ | $0.4169 \pm 0.0003$ |
| d.1 | $pp \to W^+W^-$ | $2m_W$ | 4 | $29.976 \pm 0.004$ | $43.92 \pm 0.03$ |
| d.2 | $pp \to W^+W^-\, j$ | $2m_W$ | 4 | $11.613 \pm 0.002$ | $15.174 \pm 0.008$ |
| d.3 | $pp \to W^+W^+\, jj$ | $2m_W$ | 4 | $0.07048 \pm 0.00004$ | $0.1377 \pm 0.0005$ |
| e.1 | $pp \to HW^+$ | $m_W + m_H$ | 5 | $0.3428 \pm 0.0003$ | $0.4455 \pm 0.0003$ |
| e.2 | $pp \to HW^+\, j$ | $m_W + m_H$ | 5 | $0.1223 \pm 0.0001$ | $0.1501 \pm 0.0002$ |
| e.3 | $pp \to HZ$ | $m_Z + m_H$ | 5 | $0.2781 \pm 0.0001$ | $0.3659 \pm 0.0002$ |
| e.4 | $pp \to HZ\, j$ | $m_Z + m_H$ | 5 | $0.0988 \pm 0.0001$ | $0.1237 \pm 0.0001$ |
| e.5 | $pp \to Ht\bar{t}$ | $m_{top} + m_H$ | 5 | $0.08896 \pm 0.00001$ | $0.09869 \pm 0.00003$ |
| e.6 | $pp \to Hb\bar{b}$ | $m_b + m_H$ | 4 | $0.16510 \pm 0.00009$ | $0.2099 \pm 0.0006$ |
| e.7 | $pp \to Hjj$ | $m_H$ | 5 | $1.104 \pm 0.002$ | $1.036 \pm 0.002$ |

# Summary

- One-loop integrals can be written as coefficients a, b, c and d times scalar functions and a rational part *R*
- The traditional approach for computing one-loop diagrams (Passarino-Veltman reduction) becomes more and more complicated and difficult to automate when the number of external particles increases
- The OPP reduction works at the integrand level: choosing specific values of the loop momentum results in a linear system of equations, which can be solved numerically and efficiently
- MadGraph has been extended to compute loops by using the OPP reduction as implemented in the CutTools computer code
- MadLoop generates loop diagrams by cutting them open, which results in tree-level diagrams with two extra external particles
- CutTools provides the values for which the numerator should be computed numerically and solves the resulting system of equations numerically