Automation of the FKS subtraction in MadFKS

Rikkert Frederix
University of Zurich

in collaboration with
Stefano Frixione, Fabio Maltoni & Tim Stelzer

JHEP **0910** (2009) 003 [arXiv:0908.4272 [hep-ph]]

HO10, CERN, June 21 - July 9, 2010

Next-to-leading order

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

'Real emission' NLO corrections

'Virtual' or 'one-loop' NLO corrections

'Born' or 'LO' contribution

Why automate?

- To save time

  NLO calculations can take a long time. It would be nice to spend this time doing phenomenology instead.

- To reduce the number of bugs in the calculation

  Having a code that does everything automatically will be without* bugs once the internal algorithms have been checked properly.

- To have all processes within one framework

  To learn how to use a new code for each process is not something all our (experimental) colleagues are willing to do.

Automation of virtual corrections

- BlackHat
  Berger, Bern, Dixon, Febres Cordero, Forde, Ita, Kosower & Maitre
- Rocket
  Ellis, Giele, Kunszt, Melnikov, Schulze & Zanderighi
- Cuttools (in Helac-1Loop)
  Ossola, Papadopoulos & Pittau (& Van Hameren)
- Samurai
  Mastrolia, Ossola, Reiter & Tramontano
- Golem
  Binoth, Guffanti, Guillet, Heinrich, Karg, Kauer, Pilon, Reiter & Sanguinetti
- and many others...
  Lazopoulous, Kilian, Kleinschmidt, Winter, Denner, Dittmaier, Pozzorini...

IR DIVERGENCE

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

- Real emission -> IR divergent
- (UV-renormalized) virtual corrections -> IR divergent
  - After integration, the sum of all contributions is finite (for infrared-safe observables)
  - To see this cancellation the integration is done in a non-integer number of dimensions: Not possible with a Monte-Carlo integration

Subtraction terms

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

Subtraction terms

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

$$\sigma^{\rm NLO} = \int_{m+1} \left[d^{(4)}\sigma^R - d^{(4)}\sigma^A\right] + \int_m \left[d^{(4)}\sigma^B + \int_{\rm loop} d^{(d)}\sigma^V + \int_1 d^{(d)}\sigma^A\right]_{\epsilon=0}$$

- Include subtraction terms to make real emission and virtual contributions separately finite

- All can be integrated numerically

Automation of subtraction schemes

- Catani-Seymour dipole subtraction *Catani & Seymour 1997; Phaf, Weinzierl 2001; Catani, Dittmaier, Seymour & Trocsanyi 2002*
  - implementations by various groups
    Gleisberg & Krauss;
    Seymour & Tevlin;
    RF, Gehrmann & Greiner;
    Hasegawa, Moch & Uwer;
    Czakon, Papadopoulos & Worek
- FKS subtraction *Frixione, Kunzst & Signer 1996*
  - implemented in MadFKS *RF, Frixione, Maltoni & Stelzer* and the POWHEG BOX *Alioli, Nason, Oleari & Re*

FKS subtraction

- FKS subtraction: Frixione, Kunszt & Signer 1996. Standard subtraction method in MC@NLO and POWHEG, but can also be used for 'normal' NLO computations
- Also known as "residue subtraction"
- Based on using plus-distributions to regulate the infrared divergences of the real emission matrix elements

FKS FOR BEGINNERS

- Easiest to understand by starting from real emission:

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

- $|M^{n+1}|^2$ blows up like $\dfrac{1}{\xi_i^2}\dfrac{1}{1-y_{ij}}$ with $\begin{array}{l}\xi_i = E_i/\sqrt{\hat{s}} \\ y_{ij} = \cos\theta_{ij}\end{array}$

- 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

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

FKS for beginners

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

- Definition plus distribution

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

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

FKS for beginners

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

- Definition plus distribution

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

- 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

$$\sigma^{\rm NLO} = \int_{m+1} \left[d^{(4)}\sigma^R - d^{(4)}\sigma^A \right] + \int_m \left[d^{(4)}\sigma^B + \int_{\rm loop} d^{(d)}\sigma^V + \int_1 d^{(d)}\sigma^A \right]_{\epsilon=0}$$

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

FKS -- TECHNICALITIES

- No need to change anything for BSM physics. Massive particles have only soft singularity which is independent of the spin
- Each phase space partition can be run completely independently of all the others -> genuine parallelization, i.e. with different phase-space parameterizations
- Naive scaling of the number of subtraction terms is n^2 (as opposed to n^3 of CS dipoles). Can be greatly reduced by using symmetry of the matrix elements
    - Adding additional gluons does not lead to more phase-space partitions
- In a given phase space partition, Born amplitudes need be computed only once for each real-emission event, and can be used for the Born and collinear, soft and soft-collinear counter events (and their remainders)

MadFKS

- Automatic FKS subtraction for QCD within the MadGraph/MadEvent framework
- Given the (n+1) process, it generates the real, all the subtraction terms and the Born processes
- For a NLO computation, only the finite parts of the virtual corrections are needed from the user

MadFKS -- technicalities

- Completely general & all automatic
- Same user-friendly interface as MadGraph
- MadFKS works also for any BSM physics model implemented in MadGraph, e.g. MSSM
- Color-linked Borns generated by MadDipole *RF, Gehrmann & Greiner*
- MC-ing over helicities possible; only more efficient for high-multiplicity final states
- Phase-space generation for the (n)-body is the same as in standard MG. It has been heavily adapted to generate (n+1)-body emission events at the same time
- Phase-space integration deals with the (n) and (n+1)-body processes at the same time, or separately

Full NLO

- Of course, to get the total NLO results, the finite parts of the virtual corrections should be included as well
- Interface to link with the virtual corrections following the Binoth-Les Houches Accord
  - Standardized way to link MC codes to one-loop programs
- We are also working on an interface to CutTools

  In collaboration with Hirschi, Garzelli & Pittau

Binoth-Les Houches Accord

arXiv:1001.1307 [hep-ph]

- **Facilitate the information exchange between the MC codes and the One-loop Programs (OLPs)**
- It should NOT constrain the OLP (nor the MC code) in any way
  Not a standard on what kind of information*, but more on the way it should be passed.
- OLP and MC might work in completely different ways
  Amplitudes may be created on the fly, or read from a library of processes

"Dedicated to the memory of, and in tribute to, Thomas Binoth, who led the effort to develop this proposal for Les Houches 2009"

The advantages

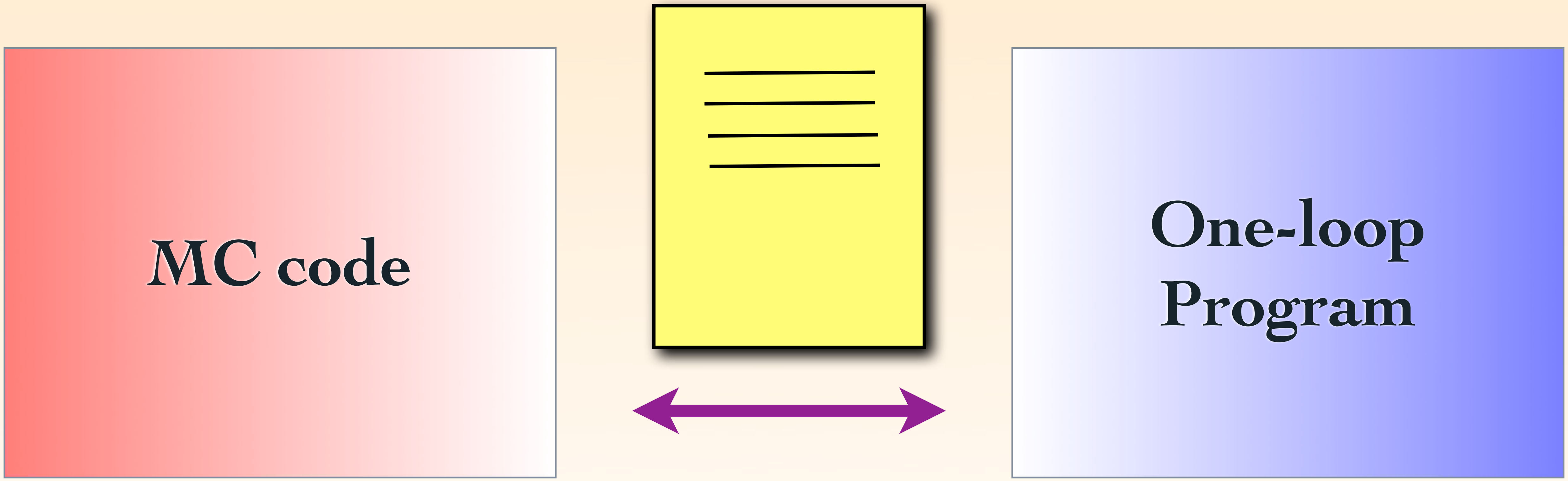
- Switching between codes becomes easy
  Model parameters etc. should be set automatically: checking codes becomes much simpler
- If you write your own OLP or MC code, you know how to link it to existing codes
  Modular problem/calculation allows for modular solutions
- Our (experimental) colleagues can still use their favorite MC code (e.g. Sherpa or MG/ME), but then at NLO, using the most efficient OLP

Binoth-Les Houches Accord

- **Initialization phase**
  MC code communicates basic information about the process to the OLP. OLP answers if it can provide the loop corrections.

- **Run-time phase**
  MC code queries the OLP for the value of the one-loop contributions for each phase-space point.

Initialization phase



MC code writes an order file
OLP replies with a contract file

```
# example order file

MatrixElementSquareType CHsummed
IRregularisation        CDR
OperationMode           LeadingColor
ModelFile               ModelInLHFormat.slh
SubdivideSubprocess     yes
AlphasPower             3
CorrectionType          QCD

#g g  ->  t tbar g
  21 21 -> 6 -6 21
#u ubar  ->  t tbar g
  2 -2 -> 6 -6 21
#u g  ->  t tbar u
  2 21 -> 6 -6 2
```

MC code

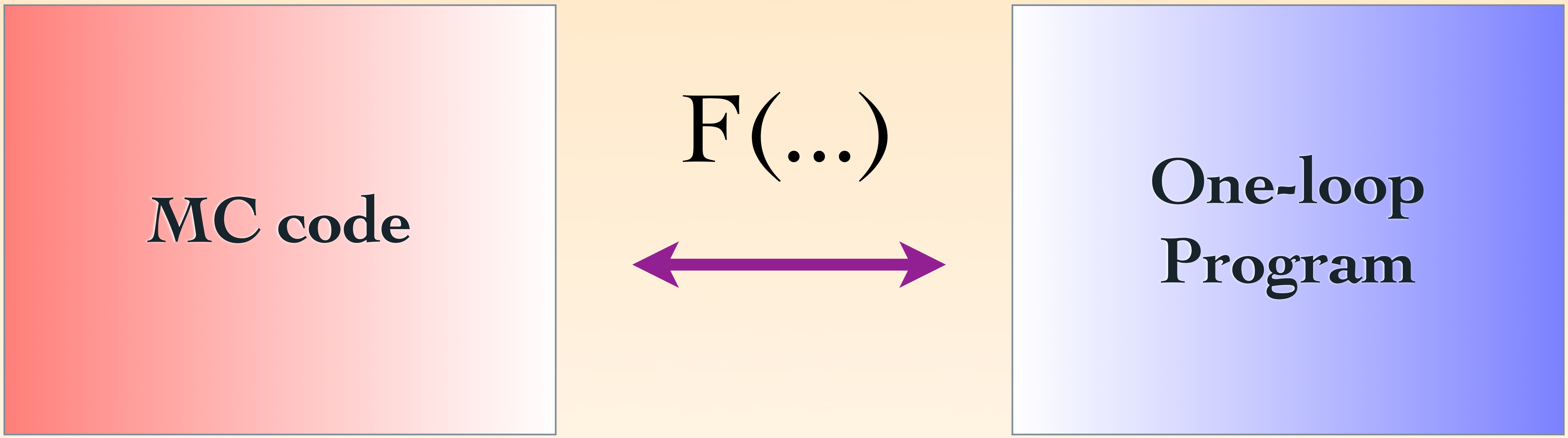
OLP

```
# example contract file
# authors of OLP, citation policy, etc

MatrixElementSquareType CHsummed              | OK
IRregularisation        CDR                   | OK
OperationMode           LeadingColor          | OK
ModelFile               ModelInLHFormat.slh   | OK
SubdivideSubprocess     yes                   | OK
AlphasPower             3                     | OK
CorrectionType          QCD                   | OK

#g g   ->  t tbar g
  21 21 -> 6 -6 21      | 2 13 35
#u ubar   ->  t tbar g
  2 -2 -> 6 -6 21       | 1 29
#u g   ->  t tbar u
  2 21 -> 6 -6 2        | 3 8 23 57
```

Run-time phase




```
OLP_Start(..)

OLP_EvalSubProcess(..)
```

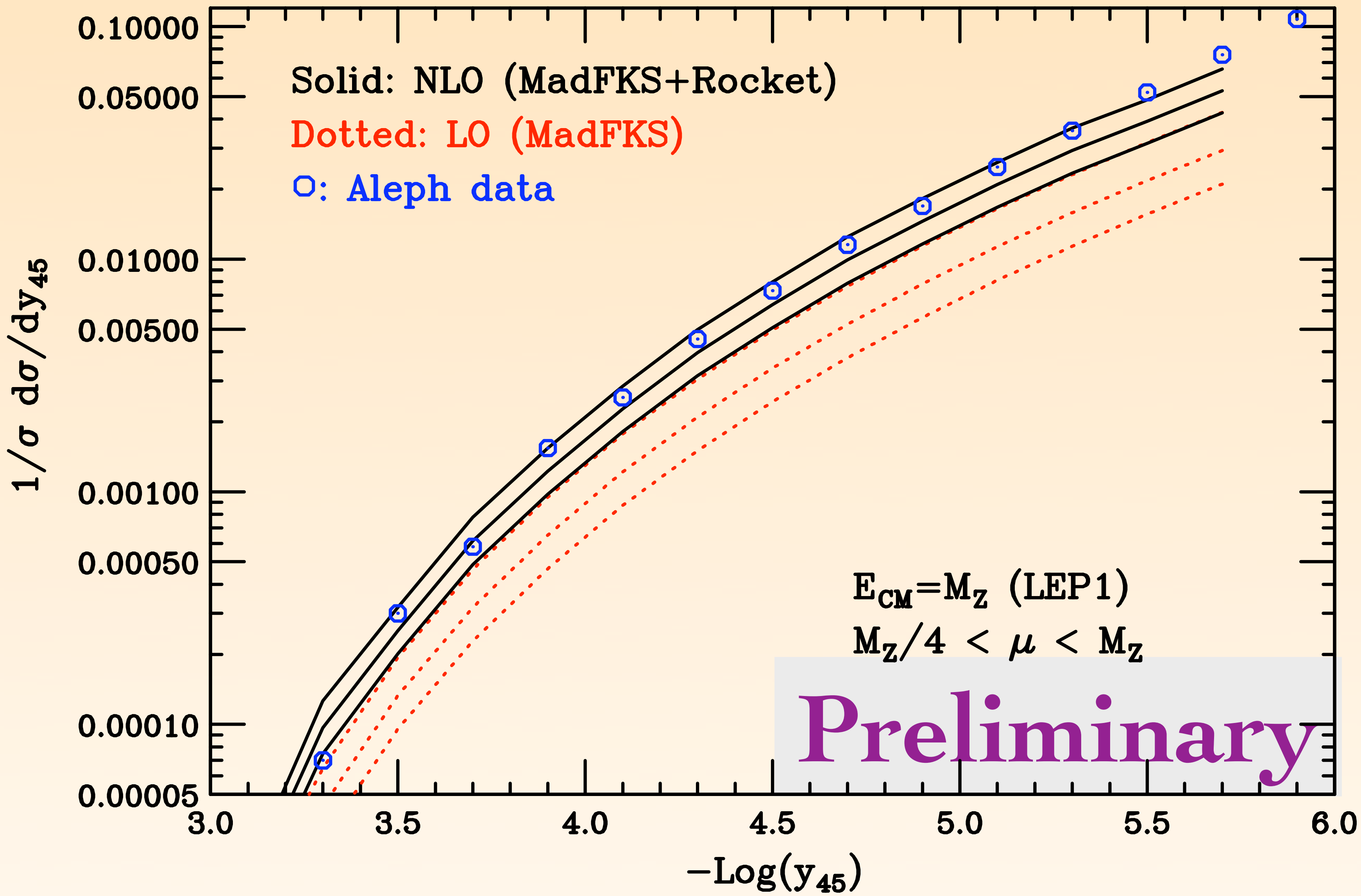
OLP_Start(..)

- Should be called once (from MC code) at start up, to confirm the contract and initialize the process
- Two arguments:
  - String with the location of the agreed contract file
  - OLP returns with integer: ‘1’ if all okay, ‘0’ if some error occurred

OLP_EvalSubProcess(..)

- Should be called (from MC code) for every phase-space point
- Five arguments:
  - Integer label of the process
  - Array of momenta and masses of the particles
  - Renormalization scale
  - Strong coupling at the renormalization scale
  - OLP returns array of the results

5 JETS AT LEP1 @ NLO



RF, Frixione, Melnikov, Stenzel, Zanderighi

- Scale dependence: +45% -30% at LO; ±20% at NLO
- Rocket and BlackHat agree pointwise
- Observable not ideal for fixed-order calculations; α_s fit is not competitive

Ongoing work on MadFKS

- Working out a version of the FKS subtraction organized as a systematic expansion in $1/N_C$ that is easy to implement in MadFKS

- We may want to integrate topologically similar subprocesses simultaneously

- Automatic MC@NLO
in collaboration with Torrielli

Automation of MC@NLO

$$d\sigma^{(\mathbb{H})}_{\text{MC@NLO}} = d\phi_{n+1}\Big(\mathcal{M}^{(r)}(\phi_{n+1}) - \mathcal{M}^{(\text{MC})}(\phi_{n+1})\Big)$$

$$d\sigma^{(\mathbb{S})}_{\text{MC@NLO}} = \int_{+1} d\phi_{n+1}\Big(\mathcal{M}^{(b+v+rem)}(\phi_n) - \mathcal{M}^{(c.t.)}(\phi_{n+1}) + \mathcal{M}^{(\text{MC})}(\phi_{n+1})\Big)$$

- In black: pure NLO, fully tested in MadFKS
- In red: already implemented (for Herwig 6), and is being tested
  - FKS is based on a collinear picture, so are the MC counter terms: branching structure is for free
  - Automatic determination of color partners
  - Automatic computation of leading-color matrix elements
  - Works also when MC-ing over helicities

To conclude

- For any QCD NLO computation (SM & BSM) MadFKS takes care of:
  - Generating the Born, real emission, subtraction terms, phase-space integration and overall management of symmetry factors, subprocess combination etc.
- External program(s) needed for the (finite part of the) loop contributions (so far working with BlackHat and Rocket)
  - BLH-interface: other codes/groups more than welcome!
- With the shower subtraction terms, interface to showers to generate automatically unweighted events with NLO precision is in testing phase