Speed gains in MadGraph5_aMC@NLO

Rikkert Frederix
CERN theory group

2nd Fast Monte Carlo Workshop in HEP

New code released

- MadGraph5 and aMC@NLO have joined forces. Last December's release of the code contains both MadGraph5 and aMC@NLO and is therefore called:

MadGraph5_aMC@NLO_v2.0.0

- It has all the current features of MadGraph5 and aMC@NLO, in addition to some new ones
- This code focusses solely on the hard event generation. Parton showering and all that is done by matching to pythia6/8 or Herwig(++)

MadGraph5_aMC@NLO example

	LO	NLO
Start the python interface	./bin/mg5	./bin/mg5
Generate a process	generate p p > t t~ w+	generate p p > t t~ w+ [QCD]
write the process to disk	output MY_TTW_PROC	output MY_TTW_NLOPROC
start the event generation	launch unweighted events	launch unweighted events up to a sign
Accuracy	LO + PS	NLO + PS

Example timing profile for WJ production

```
Inclusive timing profile :
  Overall slowest channel          0:01:00 (P0_gu_wpd/GF1 [step 1])
  Average channel running time     0:00:28
  Aggregated total running time    0:22:25
Timing profile for <IS_evaluation> :
  Overall fraction of time         23.260 %
  Largest fraction of time         35.573 % (P0_udx_wpg/GF1 [step 2])
  Smallest fraction of time        13.927 % (P0_gu_wpd/GF1 [step 0])
Timing profile for <clustering> :
  Overall fraction of time         20.842 %
  Largest fraction of time         24.926 % (P0_dxu_wpg/GF2 [step 1])
  Smallest fraction of time        15.307 % (P0_ug_wpd/GF1 [step 0])
Timing profile for <PDF_Engine> :
  Overall fraction of time         19.418 %
  Largest fraction of time         25.567 % (P0_ug_wpd/GF1 [step 1])
  Smallest fraction of time        10.179 % (P0_udx_wpg/GF1 [step 0])
Timing profile for <other_tasks> :
  Overall fraction of time         15.625 %
  Largest fraction of time         25.076 % (P0_gdx_wpux/GF1 [step 2])
  Smallest fraction of time        7.992 % (P0_dxu_wpg/GF1)
Timing profile for <Reals_evaluation> :
  Overall fraction of time         8.786 %
  Largest fraction of time         12.390 % (P0_dxu_wpg/GF2 [step 1])
  Smallest fraction of time        6.237 % (P0_ug_wpd/GF2 [step 0])
Timing profile for <OneLoop_Engine> :
  Overall fraction of time         8.401 %
  Largest fraction of time         31.729 % (P0_udx_wpg/GF2 [step 0])
  Smallest fraction of time        0.396 % (P0_ug_wpd/GF2)
Timing profile for <PS_Generation> :
  Overall fraction of time         3.669 %
  Largest fraction of time         4.657 % (P0_dxu_wpg/GF2 [step 2])
  Smallest fraction of time        2.542 % (P0_ug_wpd/GF1 [step 0])
```

- With <DEBUG>-level verbose turned on, each run gives a summary on where CPU time is spend

```
INFO:
    Summary:
    Process p p > w+ j [QCD]
    Run at p-p collider (6500 + 6500 GeV)
    Total cross-section: 5.278e+04 +- 5.0e+02 pb
    Ren. and fac. scale uncertainty: +3.6% -8.1%
    Number of events generated: 10000
    Parton shower to be used: HERWIG6
    Fraction of negative weights: 0.27
    Total running time :  6m 19s
```

Predictions at LO

How do we calculate a LO cross section for 3 jets at the LHC?

I. Identify all subprocesses (gg→ggg, qg→qgg....) in:

$$\sigma(pp \to 3j) = \sum_{ijk} \int f_i(x_1) f_j(x_2) \hat{\sigma}(ij \to k_1 k_2 k_3)$$

easy

II. For each one, calculate the amplitude:

$$\mathcal{A}(\{p\},\{h\},\{c\}) = \sum_i D_i$$

difficult

III. Square the amplitude, sum over spins & color, integrate over the phase space (D ~ 3n)

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int d\Phi_p \sum_{h,c} |\mathcal{A}|^2$$

quite hard

Multi channel integration

Consider the integration of an amplitude $|M|^2$ at tree level which lots of diagrams contribute to. Let's decompose the integrand into

$$f = \sum_{i=1}^{n} f_i \quad \text{with} \quad f_i \geq 0\,, \quad \forall\ i$$

such that:

1. we know how to integrate each one of them,
2. they describe all possible peaks

$$I = \int d\vec{\Phi}\, f(\vec{\Phi}) = \sum_{i=1}^{n} \int d\vec{\Phi}\, p_i(\vec{\Phi})\, \frac{f_i(\vec{\Phi})}{p_i(\vec{\Phi})} = \sum_{i=1}^{n} I_i$$

In MG5_aMC we have $f_i = \dfrac{|A_i|^2}{\sum_j |A_j|^2} |A_{\text{tot}}|^2$

Each contribution is a separate integral
(it is computed completely independently from all the others!)

NLO corrections

- NLO corrections have three parts:
  - The Born contribution, i.e. the Leading order.
  - 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 Virtual and Real emission have one power of α_s extra compared to the Born process

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

Virtual

- OpenLoops technique (see Maierhoefer's talk) to construct numerator function
- Reduction to scalar integrals done with OPP method as implemented in CutTools
- Scalar integrals from the OneLOop package
- The virtuals are computed twice for each PS point (with original and rotated momenta) as a check of the numerical precision
  - For numerically unstable points, switch to quadruple precision (~1% of the PS points; quad precision about 100 times slower than double precision)
- In MG5_aMC beta versions: separate integration channels for the virtual corrections. In current release, combined with the rest of the integrand (see later)

Dealing with IR divergences

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

- Only finite in d-dimensions. Cannot do numerical PS integrals in d-dimensions, hence modify the integral in the following way (subtraction method):

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

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

FKS subtraction: Phase-space partitions

- Easiest to understand by starting from real emission:

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

- $|M^{n+1}|^2$ blows up like $\dfrac{1}{\xi_i^2}\dfrac{1}{1-y_{ij}}$ with

$$\xi_i = E_i/\sqrt{\hat{s}}$$

$$y_{ij} = \cos\theta_{ij}$$

- 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

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

FKS subtraction: 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: $\xi_i \longrightarrow 0$
  - Collinear: $y_{ij} \longrightarrow 1$
  - Soft-collinear: $\xi_i \longrightarrow 0 \qquad y_{ij} \longrightarrow 1$

Together with Born

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

- Each element of this sum ('FKS configuration') can be computed completely independently from all the others: trivial to have multi-channel integration based on Born diagrams augmented with sampling of the soft/collinear structure of the FKS configuration
- The soft counterterm has the same kinematics as the Born: they can be computed together
- Naively: number of integration channels equal to Born diagrams times number of FKS configurations (original MadFKS)
- Better: start from Born, and MC sum over contributing FKS configurations (since v2.0.0beta1)

NLO+PS HARD EVENT GENERATION

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

- From a computation point of view, the same equation holds, apart that
  - FKS subtraction terms are replaced by the shower subtraction terms
  - Full n+1-body PS dependence in G
  - Integral over 1-body PS cannot be done analytically
- Possible to unweight the events up to a sign
- Negatively weighted events

Separate integration channels

- All these integration channels can be integrated completely independently from any others: trivially parallelized
  - However, the more the integral is split, the larger the fraction of negatively weighted events
  - All channels (even the very small ones) need to be computed with a minimum precision before they can be discarded. Important when number of channels is larger than the number of CPU cores available
- Original version had separate channels based on
  - real emission flavour structures. Now there is a MC sum over FKS configurations (since v2.0.0beta1)
  - separate channels for the virtual corrections. Now longer needed, see next slides (since v2.0.0)

Speed improvements in v2.0.0

- MC over helicities for the virtual corrections
- "Virtual Tricks"

MC over helicities for the virtual

- Naive Monte-Carlo over helicities (i.e. randomly picking a single helicity per PS point) does not improve the speed of the code:
  - Fluctuations between PS points are greatly enhanced
  - Even including grids to pick the helicity configuration that give large contributions does not help much, because it doesn't factorize over the phase-space
- Solution: pick helicity at random for each PS point with probability based on the Born helicity configurations!
  - It's impossible that $|B_i|^2$ is zero, while $2\mathrm{Re}[V_i\, B_i^*]$ is not
  - Fluctuations between PS points not larger than when explicitly summing over helicities

“Virtual Tricks I”

- Virtuals are small
  - Dynamically determine the fraction of PS points for which to include the virtual corrections
    - when including them, their weight is multiplied by the inverse of this fraction
  - Make sure that the integration uncertainty from the virtuals alone is always smaller than the uncertainty from everything else. This determines the minimal fraction

"Virtual Tricks II"

- Sometimes virtuals are not small, which means that we cannot significantly reduce the fraction of PS points for which we include them
  - Examples are Wbb, Zbb, ttbb, WWbb (It seems very much related to the 4F scheme, although single-top 4F seems an exception...)
- But we can make the virtual small!
  - Virtuals are almost equal to a constant times the Born
  - Determine this constant dynamically and include this approximation for every phase-space point, and use the "Virtual Tricks I" on the left-over difference
  - In fact, we don't use a single constant, but we setup a (coarse) grid to have some phase-space dependence in the constant

IN PRACTICE:

MADGRAPH5 2.0.0BETA3

```
Summary:
Process p p > t t~ [QCD]
Run at p-p collider (4000 + 4000 GeV)
Total cross-section: 1.770e+02 +- 1.7e+00 pb
Ren. and fac. scale uncertainty: +13.5% -13.0%
Number of events generated: 10000
Parton shower to be used: HERWIG6
Fraction of negative weights: 0.16
Total running time : 12m 12s

DEBUG:
Number of loop ME evaluations: 168120
```

MADGRAPH5_AMC@NLO

```
Summary:
Process p p > t t~ [QCD]
Run at p-p collider (4000 + 4000 GeV)
Total cross-section: 1.765e+02 +- 9.2e-01 pb
Ren. and fac. scale uncertainty: +12.2% -12.3%
Number of events generated: 10000
Parton shower to be used: HERWIG6
Fraction of negative weights: 0.16
Total running time : 1m 35s

DEBUG:
Number of loop ME evaluations (by MadLoop): 6967
```

In practice:

Madgraph5 2.0.0beta3

```
Summary:
Process p p > t t~ [QCD]
Run at p-p collider (4000 + 4000 GeV)
Total cross-section: 1.770e+02 +- 1.7e+00 pb
Ren. and fac. scale uncertainty: +13.5% -13.0%
Number of events generated: 10000
Parton shower to be used: HERWIG6
Fraction of negative weights: 0.16
Total running time : 12m 12s

DEBUG:
    Number of loop ME evaluations: 168120
```

N.A.

MadGraph5_aMC@NLO

```
Summary:
Process p p > t t~ [QCD]
Run at p-p collider (4000 + 4000 GeV)
Total cross-section: 1.765e+02 +- 9.2e-01 pb
Ren. and fac. scale uncertainty: +12.2% -12.3%
Number of events generated: 10000
Parton shower to be used: HERWIG6
Fraction of negative weights: 0.16
Total running time : 1m 35s

DEBUG:
    Number of loop ME evaluations (by MadLoop): 6967
```

```
Summary:
Process p p > t b t~ b~ [QCD]
Run at p-p collider (4000 + 4000 GeV)
Total cross-section: 2.671e+00 +- 1.2e-02 pb
Ren. and fac. scale uncertainty: +39.1% -27.8%
Number of events generated: 200000
Parton shower to be used: HERWIG6
Fraction of negative weights: 0.29
Total running time :  17h 0m
Sequential running time : ~ 6 days

DEBUG:
    Number of loop ME evaluations (by MadLoop): 367802
```

2.6s per unweighted event on a single CPU core

Theory uncertainties

- A great advantage of using NLO computations is that theory uncertainties can easily be estimated by
  - varying renormalisation and factorisation scales (9 values)
  - evaluating the PDF error sets (40-100 values)
- In principle each scale combination and PDF error set requires a separate run.
- However, the only dependence is in the hard events (in a very good approximation): instead of generating ~50 sets of events that are all unweighted, generate 1 set of events where each event has ~50 weights
- Other uncertainties, intrinsic to the parton shower, (e.g. resummation scale or 'Herwig vs. Pythia') do require several runs because they will give different particle configurations

HEADER

```
<header>
 ...

 <initrwgt>
  <weight id='1'> This is the original event weight </weight>
  <weightgroup type='scale_variation' combine='envelope'>
     <weight id='2'> muR=2.0 </weight>
     <weight id='3'> muR=0.5 </weight>
  </weightgroup>
   <weightgroup type="mrst2008e40" combine="hessian">
     <weight id='4'> set01 </weight>
     <weight id='5'> set02 </weight>
     ...
   </weightgroup>
  <weightgroup type='Qmatch_variation' combine='envelope'>
     <weight id='44'> Qmatch=20 </weight>
     <weight id='45'> Qmatch=40 </weight>
  </weightgroup>
  <weight id='46'> BSM benchmark point number 42B, see arXiv XXXX.XXXX </weight>
 </initrwgt>
...
</header>
```

Each event

```
<event id='evtid'>
7 100  0.10000000E+01  0.20000000E+00  0.00000000E+00  0.00000000E+00
 -2 -1  0  0 0 0  0.12699952E+01  0.55429630E+01  0.57634577E+02  0.57914435E+02  0.000
  2 -1  0  0 0 0 -0.91353745E+00  0.13160013E+01 -0.34965448E+02  0.35002128E+02  0.000
 23  2  1  1 0 0  0.35645919E+00  0.68589662E+01  0.22669189E+02  0.92916566E+02  0.898
-13  2  3  3 0 0  0.51612833E+01  0.21143065E+02  0.53960893E+02  0.58184682E+02  0.105
 13  2  3  3 0 0 -0.48048241E+01 -0.14284099E+02 -0.31291705E+02  0.34731884E+02  0.105
-13  1  0  0 0 0  0.51612833E+01  0.21143065E+02  0.53960893E+02  0.58184682E+02  0.105
 13  1  0  0 0 0 -0.48048241E+01 -0.14284099E+02 -0.31291705E+02  0.34731884E+02  0.105
 <rwgt>
  <wgt id='1'> 1.001e+00 </wgt>
  <wgt id='2'> 0.204e+00 </wgt>
  <wgt id='3'> 1.564e+00 </wgt>
  <wgt id='4'> 2.248e+00 </wgt>
  <wgt id='5'> 1.486e+00 </wgt>
  ...
  <wgt id='46'> -0.899e+00 </wgt>
 </rwgt>
</event>
```

LO reweighting

- Currently, when running MadGraph5_aMC@NLO at NLO only scale and PDF uncertainties can be generated in this way
    - When running LO only, also model parameters can be varied. This is useful for e.g. BSM parameter scans or the generation of templates used in the W-boson mass measurement

MLM & FxFx merging scale

- One of the advantages of using MLM matching (or FxFx matching at NLO) is that the hard matrix elements do not depend on the merging scale. (There is only a lower cut that should be always well-below this scale). The merging scale enters only in the rejection of the events after showering. This means that a single event sample can be generated where different fractions of events are included or not depending on the merging scale: easy to assess systematics due to merging scale

Further improvements?

- Phase-space integration optimized for event generation/unweighting
  - Unweighting efficiency usually better than 0.1% (for 2->4 NLO processes)
  - Not so good for fixed order runs; possible to optimize better
- Possible to reduce the number of integration channels quite a bit more by 'grouping' flavour combinations: many Born topologies are shared between many flavour combinations. Already done for LO runs

Conclusions

- Performing NLO+PS computations has become as easy as LO+PS computations from the user point of view
  - However, technically they are a lot more complicated and CPU intensive
- Still, constantly improvements are being found
  - OpenLoops
  - Virtual tricks
  - Reweighting to get theory uncertainty bands
- For simple processes dominated by FastJet & LHAPDF:
  only way to improve is to increase the unweighting efficiency with smarter phase-space mappings, reducing the overall number of PS points needed