Towards automated NLO matching

Ken Arnold

16.2.2012

What is NLO matching?

Observable O at NLO:

$$\langle O\rangle_{NLO} = O(0)\,[B+\alpha V] + \int_0^1 \mathrm{d}x\, O(x)\alpha\frac{R(x)}{x}$$

with

B : Born cross-section

V : Virtual cross-section

$\frac{R(x)}{x}$: Real differential cross-section

x : Phase space variable of additional emission;

$x = 0$ for no emission

α : Coupling constant

Divergencies in V cancel against those stemming from the integration of $\frac{R(x)}{x}$.
$\rightarrow$ Do subraction to render both parts seperately finite.

Subtraction formalism:

$$\langle O\rangle_{NLO} = O(0)\left[B+\alpha V+\int_0^1 \frac{A(x)}{x}\right] + \int_0^1 \mathrm{d}x\,\alpha\left[O(x)\frac{R(x)}{x} - O(0)\frac{A(x)}{x}\right]$$

Rewrite this as

$$\langle O\rangle_{NLO} = O(0)\left[B+\alpha\tilde{V}\right] + \int_0^1 \mathrm{d}x\,\alpha\left(\frac{O(x)R(x)-O(0)A(x)}{x}\right)$$

where $\tilde{V}$ and $\int_0^1 \mathrm{d}x\,(\ldots)$ are separately finite.

Parton shower approximation with Sudakov form factor

$$\Delta(x_0, x_1) = \exp\left\{ -\int_{x_0}^{x_1} \mathrm{d}x\, \alpha \frac{P(x)}{x} \right\} \approx 1 - \int_{x_0}^{x_1} \mathrm{d}x\, \alpha \frac{P(x)}{x}$$

acting on LO calculation generates terms of order α.

$$\begin{aligned} \langle O \rangle_{PS} &= O(0) B \Delta(\mu, 1) + \int_{\mu}^{1} \mathrm{d}x\, \alpha O(x) B \frac{P(x)}{x} \Delta(\mu, x) \\ &\approx O(0) B \left[1 - \int_{\mu}^{1} \mathrm{d}x\, \alpha \frac{P(x)}{x} \right] + \int_{\mu}^{1} \mathrm{d}x\, \alpha O(x) B \frac{P(x)}{x} \end{aligned}$$

$\rightarrow$ *Double counting when acting on NLO calculation.*

Subtract terms before showering:

$$\langle O\rangle'_{NLO} = O(0)\left[B + \alpha\tilde{V}\right] + \int_0^1 \mathrm{d}x\,\alpha\left(\frac{O(x)R(x) - O(0)A(x)}{x}\right) + O(0)B\int_\mu^1 \mathrm{d}x\,\alpha\frac{P(x)}{x} - \int_\mu^1 \mathrm{d}x\,\alpha O(x)B\frac{P(x)}{x}$$

Matched observable:

$$\langle O\rangle_{MC@NLO} = O(0)\left[B + \alpha\tilde{V} + \int_0^1 \mathrm{d}x\,\alpha\frac{BP(x) - A(x)}{x}\right] + \int_0^1 \mathrm{d}x\,\alpha O(x)\frac{R(x) - BP(x)}{x}$$

$$\langle O\rangle_{MC@NLO} = O(0)\left[B + \alpha\tilde{V} + \int_0^1 \mathrm{d}x\,\alpha\frac{BP(x) - A(x)}{x}\right] + \int_0^1 \mathrm{d}x\,\alpha O(x)\frac{R(x) - BP(x)}{x}$$

- Events with Born-type and with real emission configuration are seperately finite
- Expanding the formula above with parton shower recovers correct NLO cross section
- We have dropped terms of the form

$$\int_0^{\mu} \mathrm{d}x\,\alpha\,[O(0) - O(x)]\,B\frac{P(x)}{x}$$

$$\langle O\rangle_{MC@NLO} = O(0)\left[B + \alpha\tilde{V} + \int_0^1 \mathrm{d}x\,\alpha\frac{BP(x) - A(x)}{x}\right] + \int_0^1 \mathrm{d}x\,\alpha O(x)\frac{R(x) - BP(x)}{x}$$

Possible simplifications:

- $R(x) - BP(x) = 0$: Use exact real emission ME for first PS splitting (POWHEG)
- $BP(x) - A(x) = 0$: PS uses the same splitting functions as subtraction scheme (e.g. dipole shower)

Matchbox

Matchbox [*KA, S. Gieseke, J. Kotanski, S. Plätzer, M. Stoll*]:
NLO framework within ThePEG and Herwig++

- Automated CS dipole subtraction
- Automated NLO matchings both POWHEG and MC@NLO type
- Automated phasespace generation
- Adaptive sampler for Sudakov type distributions
- User sees only the Herwig input file he is used from LO + parton shower calculations!
- External matrix elements can be interfaced in two ways: Either via squared matrix elements or via colour ordered amplitudes

Matchbox is going to be released with the next Herwig++ update!

Phasespace & Samplers

- *Phasespace generator* maps random numbers onto physical variables to construct kinematics of an event. For n outgoing particles, the phasespace of hadron collision has at least $3n-2$ dimensions. Mapping according to the peak structure of the integrand improves convergence of Monte Carlo integration.
- *Sampler* picks random numbers according to a probability distribution. This can be adapted to the integrand in various ways to sample more points in areas with a large contribution to the integral

Three possible choices for phasespace:

- RAMBO: Flat phasespace
- VBFPS: Interface to VBFNLO phasespace generator
- TreePhasespace

New built-in phasespace generator TreePhasespace:

- Uses Herwig's internal information on diagrams to map propagator invariants with the correct singular structure of each squared diagram
- Phasespace is build by sequential 1→2 and 2→2 splittings (similar to HELAC/PHEGAS)
- Multi-channel: Each diagram leads to a different phasespace mapping
- Sampler carries out adaptive channel selection, so that channels are selected according to their contribution to the cross section

Samplers:

- FlatSampler (for debugging)
- ProjectingSampler: VEGAS-like adaption to the cross-section within several iterations, but splitting bins instead of shifting borders
- ExSampler: Bisecting cells if unweighting efficiency drops below user defined value. Dimension of splitting k determined by gain measure:

$$g_k(x_k) = \frac{\int_{x_k^-}^{x_k} \mathrm{d}x \langle f \rangle_k(x) - \int_{x_k}^{x_k^+} \mathrm{d}x \langle f \rangle_k(x)}{\int_{x_k^-}^{x_k^+} \mathrm{d}x \langle f \rangle_k(x)}$$

After splitting: Presample cells to determine new maximum weights.

Goal: get unweighted events quickly!

Results

Hjj@LO

VBFNLO standalone

time	Xsec/fb	# unweighted events
1185s	3445.84 ± 0.47	368000

Matchbox: Generate 100k unweighted events with inclusive cuts

TreePS

Sampler	time read	Xsec/fb after read	time run	Xsec/fb after run	offset
ExSampler	13s	3452.9 ± 46.7	154s	3426.5 ± 19.9	-1.0
FlatSampler	15s	3452.9 ± 46.7	507s	3451.3 ± 27.3	0.2
ProjSampler	57s	3431.0 ± 14.7	—	—	—

VBFPS:

Sampler	time read	Xsec/fb after read	time run	Xsec/fb after run	offset
ExSampler	13s	3423.5 ± 123.0	516s	3317.0 ± 38.1 *	-3.4
FlatSampler	14s	3423.5 ± 123.0	5317s	3402.5 ± 71.5	-0.6
ProjSampler	50s	3370.4 ± 22.9	—	—	—

*=still compensating for new maximum

Hjjj@LO

VBFNLO standalone

time	Xsec/fb	# unweighted events
20m	769.20 ± 0.29	43000

Matchbox: Generate 100k unweighted events with inclusive cuts

TreePS, ExSampler

PreSPs	time read	Xsec/fb after read	time run	Xsec/fb after run
8k	28s	784.5 ± 51.0	33m	721.0 ± 22.5 *
32k	109s	754.1 ± 17.0	62m	741.1 ± 20.5 *

*=still compensating for new maximum

$\rightarrow$ In time per unweighted event, we still can compare with the VBFNLO standalone computation.