New algorithms to compute virtual and tree-level amplitudes

[Fermilab Theory Seminar]

Jan Winter

– Fermilab –

- *Part I – Numerical calculation of colour-dressed one-loop amplitudes* [a]
- *Part II – Graphics processing units (GPUs) for leading-order matrix-element evaluations* [b]

[a] In collaboration with: W. Giele and Z. Kunszt

[b] In collaboration with: W. Giele and G. Stavenga

Numerical calculation of

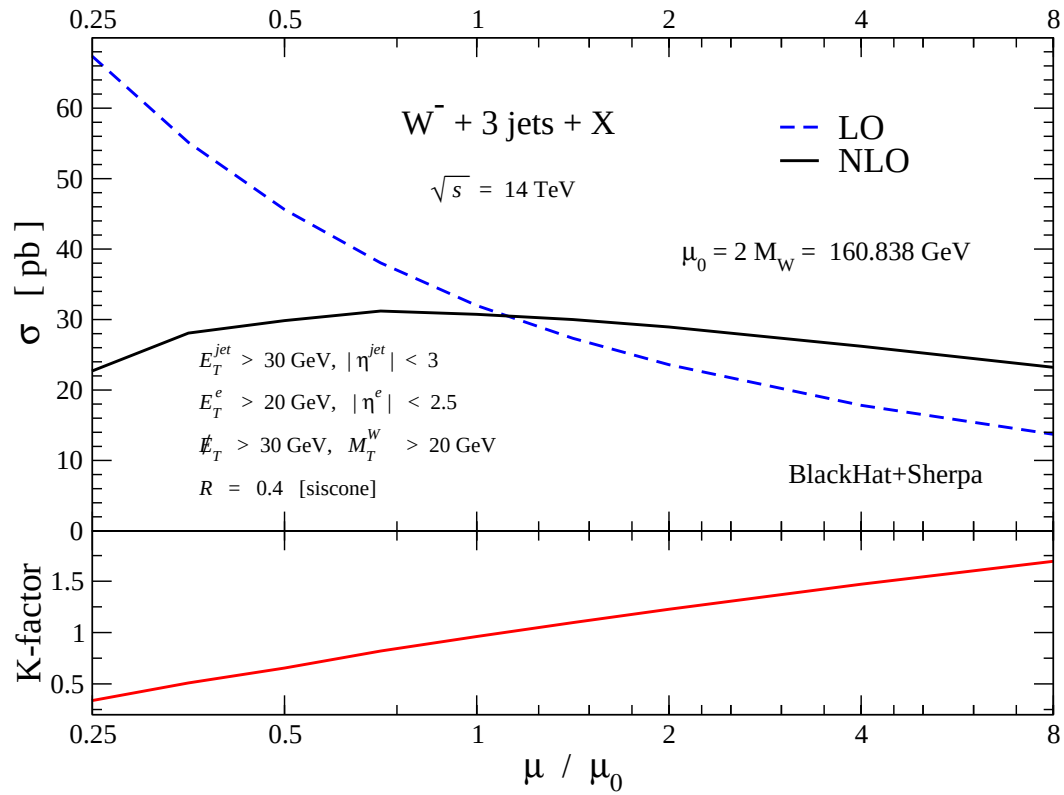
colour-dressed one-loop amplitudes

- *NLO calculations & new efforts*
- *One-loop amplitude computions & generalized unitarity*
- *Numerical method based on colour-dressing techniques*
- *Results for multiple gluon scattering*

Need for NLO calculations

➡ ***Lessons learned from LEP, HERA, Tevatron:***
LO predictions are fine, yet often only give rough estimates

@ NLO: 1st real prediction of normalization of many observables
less sensitivity to unphysical input scales ($\mu_{\rm F}$ & $\mu_{\rm R}$)
more physics (parton merging, jet substructure, ISR, more IS parton species)



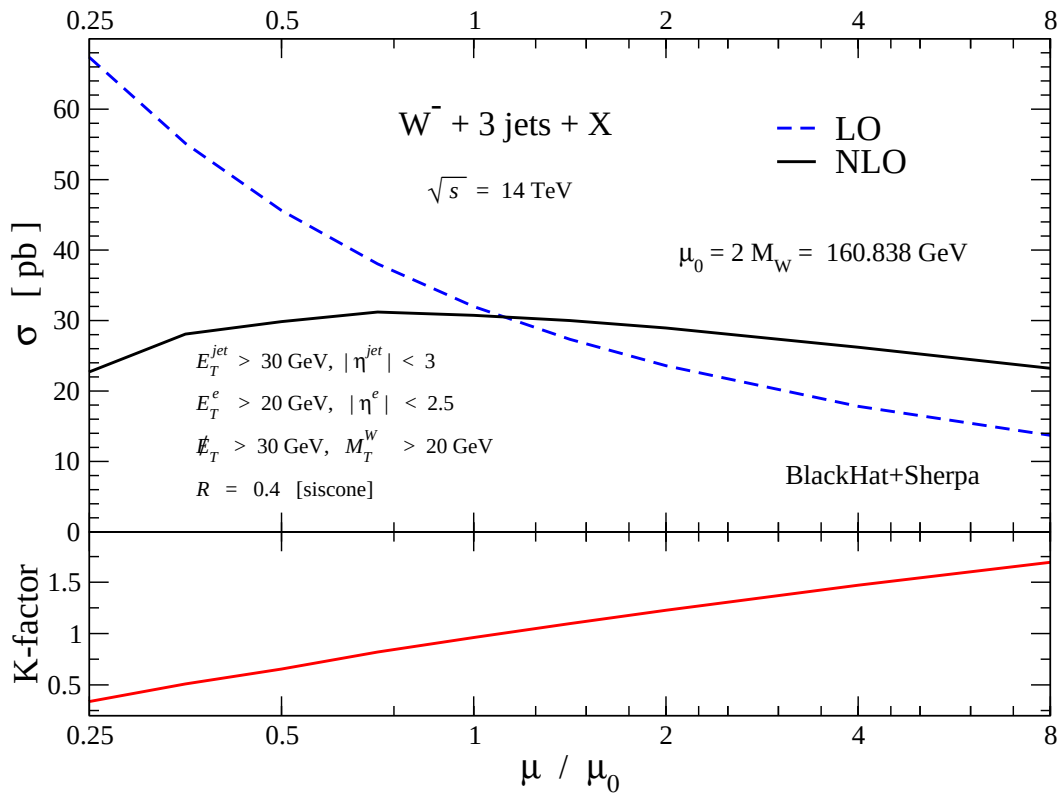
Need for NLO calculations

➡ ***Lessons learned from LEP, HERA, Tevatron:***
LO predictions are fine, yet often only give rough estimates

@ NLO: 1st real prediction of normalization of many observables
less sensitivity to unphysical input scales ($\mu_{\rm F}$ & $\mu_{\rm R}$)
more physics (parton merging, jet substructure, ISR, more IS parton species)

➡ ***Components of NLO calculations***

- tree-level amplitudes (LO & real radiation)
- \+ one-loop correction to Born level
- \+ subtraction terms to handle and combine singularities
- \+ phase-space generator



Need for NLO calculations

➡ ***Lessons learned from LEP, HERA, Tevatron:***
LO predictions are fine, yet often only give rough estimates

@ NLO: 1st real prediction of normalization of many observables
less sensitivity to unphysical input scales ($\mu_{\rm F}$ & $\mu_{\rm R}$)
more physics (parton merging, jet substructure, ISR, more IS parton species)

➡ for example, BLACKHAT+SHERPA

$$\sigma = \int_m d\sigma^B + \int_{m+1} (d\sigma^R - d\sigma^A) + \int_m (d\sigma^V + \int_1 d\sigma^A)$$

➡ ***Componen***

- tree-level a
(LO & real

\+ one-loop co

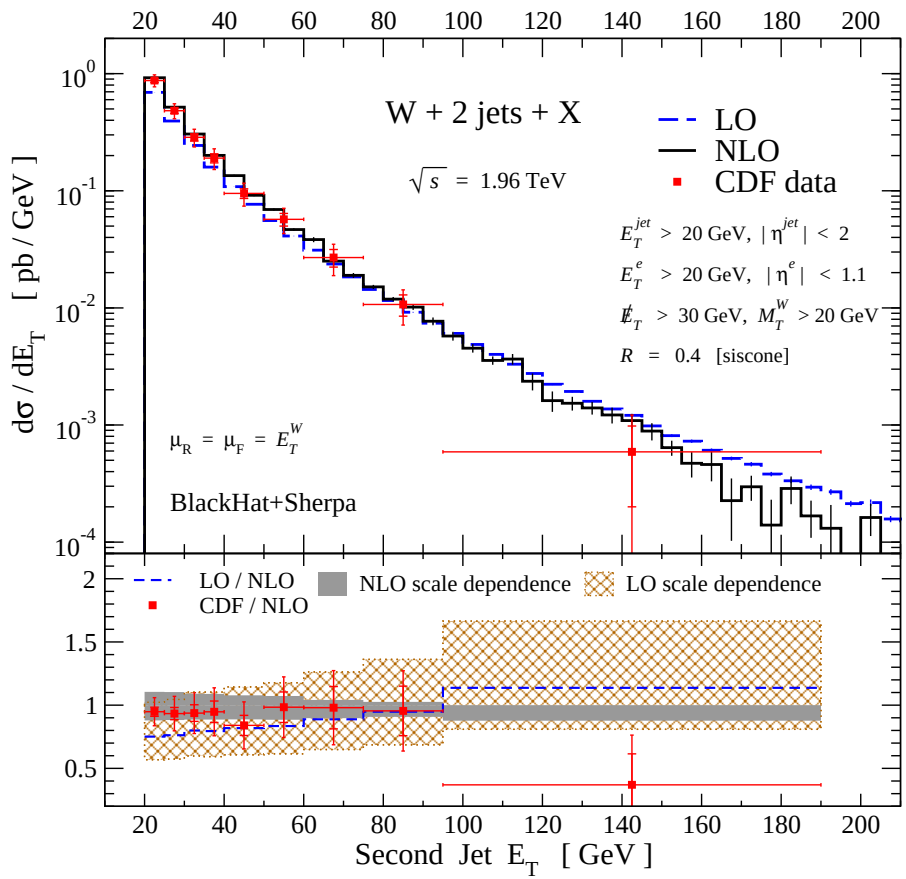
\+ subtraction
combine si

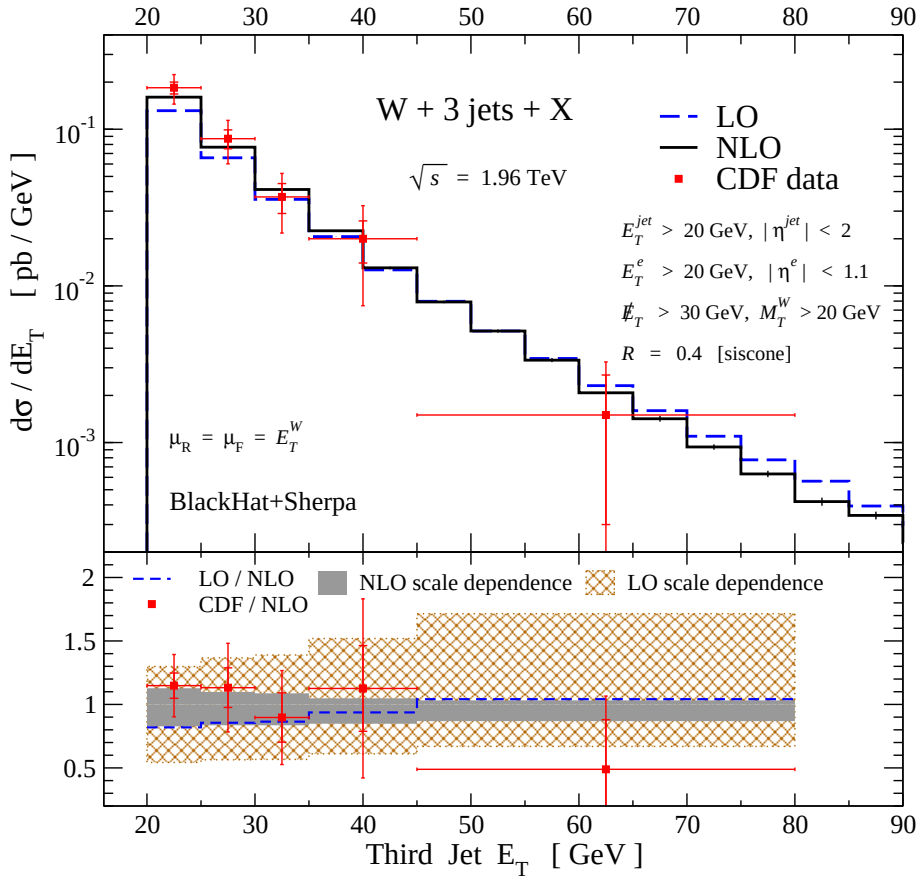
\+ phase-spac

BlackHat+Sherpa interfacing in two steps

[Gleisberg, Krauss, EPJC53 (2008) 501] [Berger et al., Phys Rev D80 (2009) 074036]

- Initialization: Process and parameter (model, μ_R scheme, sampling) agreement
- Run time: Sherpa provides phase-space point, gets back $|\mathcal{M}^{(1)}|^2$
- Interface details: Les Houches 2009 accord

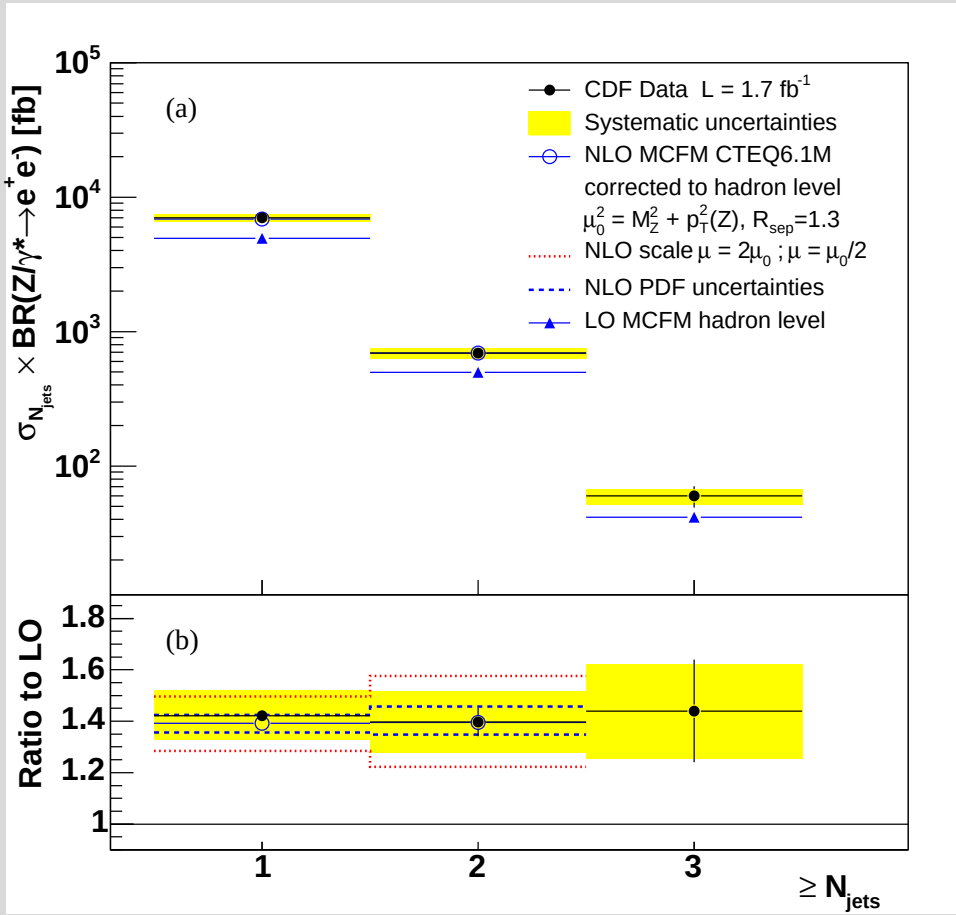


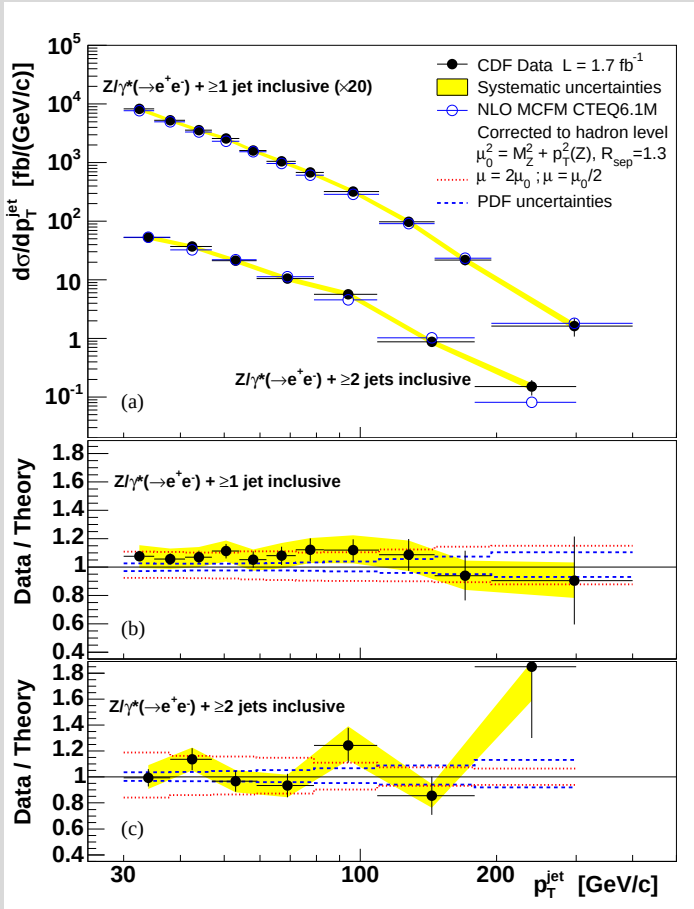


MCFM

[Campbell, Ellis, http://mcfm.fnal.gov/] [T. Aaltonen et al., PRL **100** (2008) 102001]

- NLO parton-level event generator for a variety of processes at hadron colliders.
- Everybody can run processes @ NLO (and LO) with MCFM themselves.
- Spin correlations maintained in decays. Helicity amplitudes. Slightly modified CS subtraction.



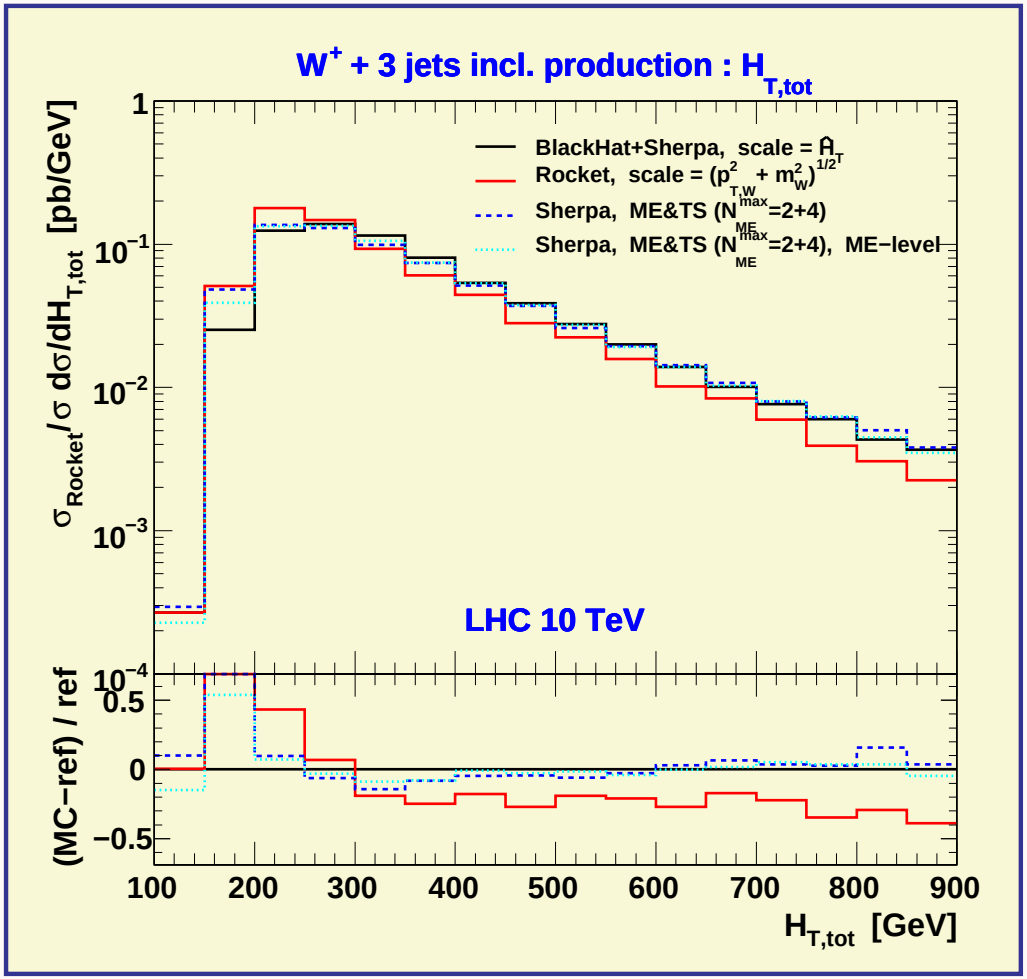


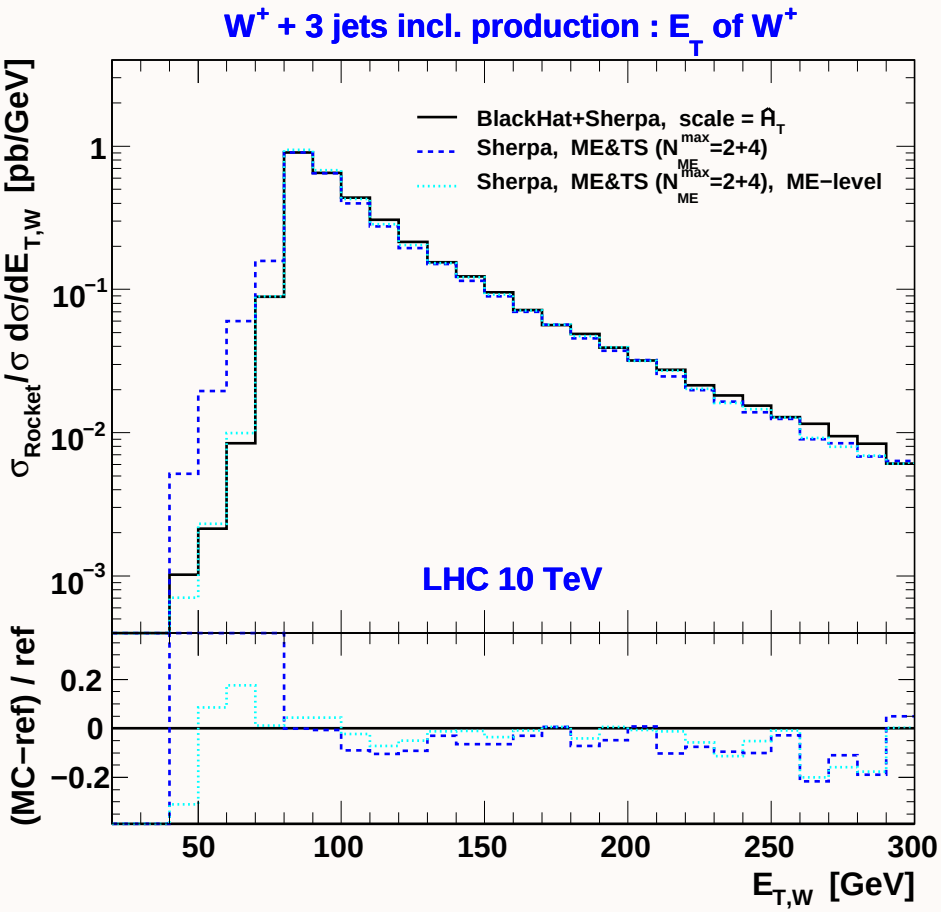
Z+jets
CDF data
2007
1.7/fb

Comparison of LHC predictions for W+3jets

[Höche, Huston, Maitre, Winter, Zanderighi; LesHouches09 proceed.: arXiv:1003.1241]

- between BlackHat [Berger et al.], Rocket [Ellis, Melnikov, Zanderighi] and Sherpa [Gleisberg et al.]
- rather different scale choices at NLO yield > 20% deviations ... impact on BSM searches !
- Sherpa vs1.2's ME&TS merging in good agreement with NLO once rescaled to NLO xsec

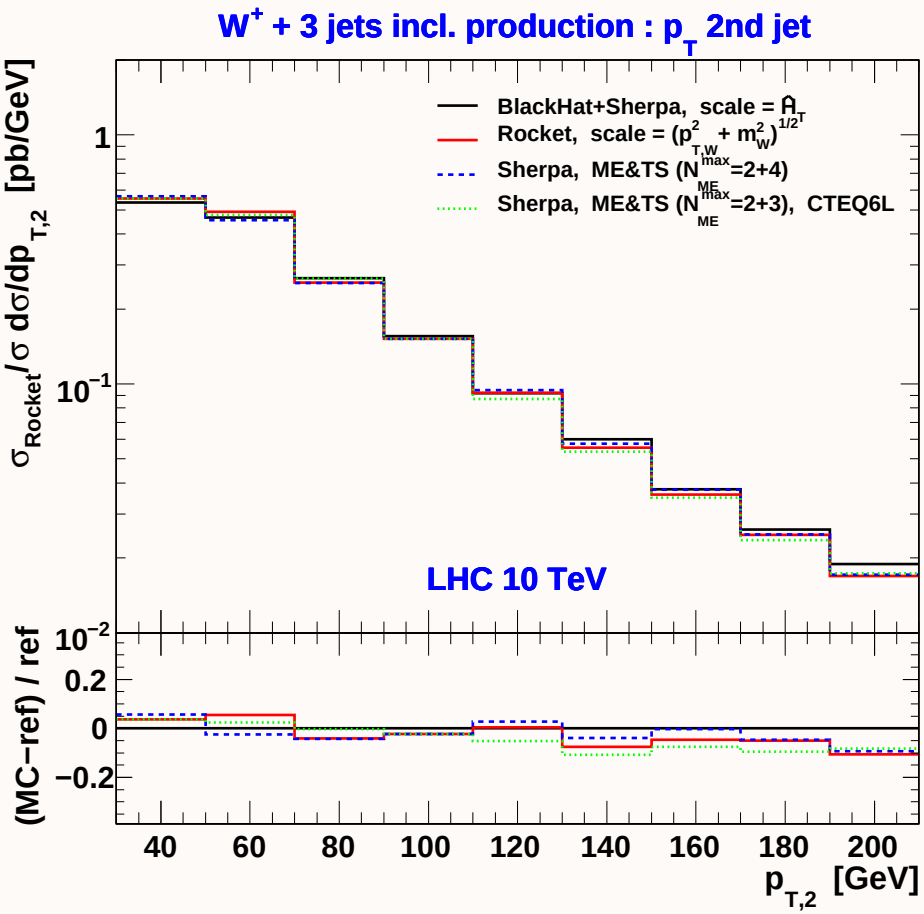


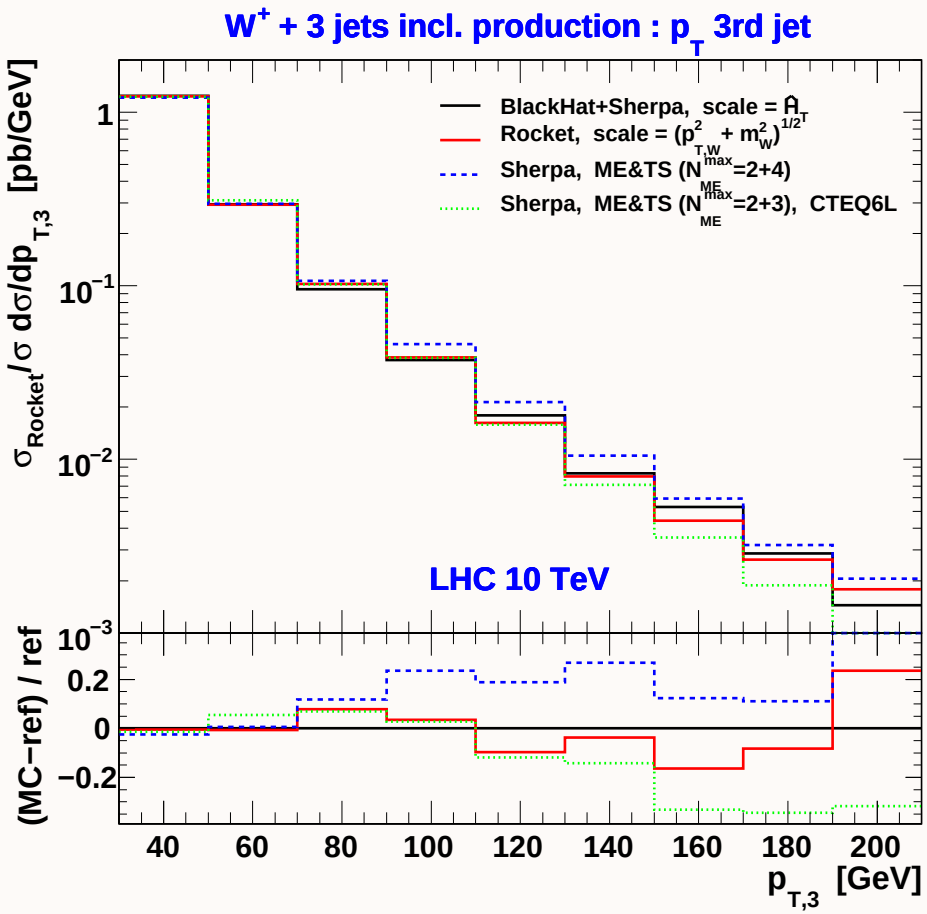


Comparison of LHC predictions for W+3jets

[Höche, Huston, Maitre, Winter, Zanderighi; LesHouches09 proceed.: arXiv:1003.1241]

- between BlackHat [Berger et al.], Rocket [Ellis, Melnikov, Zanderighi] and Sherpa [Gleisberg et al.]
- rather different scale choices at NLO yield > 20% deviations ... impact on BSM searches !
- Sherpa vs1.2's ME&TS merging in good agreement with NLO once rescaled to NLO xsec





NLO calculations

- Feynman diagram calculations: computational algorithms of at least factorial complexity
- bottleneck: virtual corrections (tensor-integral reductions generate large # of terms)
- @ tree level: algorithms of polynomial or, incl. colour, exponential complexity exist ($\tau \sim N^{\#}$ or $\#^N$)
  recursive methods efficiently re-use recurring groups of offshell Feynman graphs
- @ loop level: generalized unitarity-cut methods factorize one-loop into tree amplitudes
  computing time grows with # of cuts & depends on algorithm employed at tree level

Goal ➡ ***provide algorithm(s) [tools] of exponential complexity to calculate virtual corrections***

NLO calculations

- Feynman diagram calculations: computational algorithms of at least factorial complexity
- bottleneck: virtual corrections (tensor-integral reductions generate large # of terms)
- @ tree level: algorithms of polynomial or, incl. colour, exponential complexity exist ($\tau \sim N^{\#}$ or $\#^N$)
  recursive methods efficiently re-use recurring groups of offshell Feynman graphs
- @ loop level: generalized unitarity-cut methods factorize one-loop into tree amplitudes
  computing time grows with # of cuts & depends on algorithm employed at tree level

Goal ➡ ***provide algorithm(s) [tools] of exponential complexity to calculate virtual corrections***

➡ ***Generalized unitarity methods – active, ongoing field of research***

- Britto, Cachazo, Feng — analytic work.
- Bern, Dixon, Dunbar, Kosower — analytic work; Berger et al. — BlackHat project.
- Ossola, Papadopoulos, Pittau + Bevilacqua, Czakon, Garzelli, Hameren, Worek — CutTools/Helac-NLO.
- Ellis, Giele, Kunszt, Melnikov, Zanderighi — "Rocket Science".
- Lazopoulos — code for ordered QCD one-loop amplitudes.
- Mastrolia, Ossola, Reiter, Tramontano — Samurai.

Virtual correction and colour decomposition

$$d\sigma_{\rm V}(f_1 f_2 \to f_3 \ldots f_N) \sim \int d\Phi(p_1 \ldots p_N)\, 2\,{\rm Re}\left(\mathcal{M}^{(0)}(f_1 \ldots f_N)^* \times \mathcal{M}^{(1)}(f_1 \ldots f_N)\right)$$

➡ ***factorization of one-loop amplitude in colour factors and primitive amplitudes is systematic***

- colour decomposition of one-loop N-gluon amplitude in $SU(N_{\rm C})$ gauge theory

$$\mathcal{M}^{(1)} = g^N \sum_{\sigma \in S_{N-1}/\mathcal{R}} {\rm Tr}(F^{a_{\sigma_1}} \cdots F^{a_{\sigma_N}})\, \mathcal{A}_N^{(1)[1]}(\sigma_1, \ldots, \sigma_N) \quad +$$

$$2\, n_f\, g^N \sum_{\sigma \in S_{N-1}/\mathcal{R}} {\rm Tr}(\lambda^{a_{\sigma_1}} \cdots \lambda^{a_{\sigma_N}})\, \mathcal{A}_N^{(1)[1/2]}(\sigma_1, \ldots, \sigma_N)$$

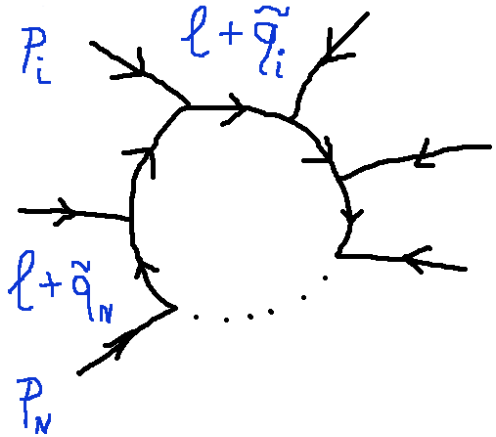
allows for separate treatment of colour factors and primitive or ordered amplitudes

- N gluons & ask for leading-colour contributions only ... make use of phase-space symmetry

$$\int d\Phi\, {\rm Re}(\mathcal{M}^{(0)*}\mathcal{M}^{(1)}) \sim \sum_{\rm perm} \int d\Phi\, {\rm Re}(\mathcal{A}^{(0)*}\mathcal{A}^{(1)}) \approx (N-1)! \int d\Phi\, {\rm Re}(\mathcal{A}^{(0)*}\mathcal{A}^{(1)})$$

➡ simplifications come in handy when calculating a specific process (both BlackHat and Rocket use these tricks) — however colour decomposition is not so optimal for automation

Decomposition of one-loop amplitudes



$$\mathcal{A}_N^{(1)}(\{p_i\}) = \int \frac{d^D\ell}{i\pi^{D/2}} \frac{\mathcal{N}(\{p_i\} \mid \ell)}{d_{i_1} d_{i_2} \cdots d_{i_N}} , \qquad d_i(\ell) = (\ell + \tilde{q}_i)^2 - m_i^2$$

➡ ***decompose into a linear sum of scalar box, triangle, bubble and tadpole master integrals (cut-constructible part) and rational terms***

$$\mathcal{A}_N^{(1)}(\{p_i\}) = \sum_{[i_1|i_4]} d_{i_1 i_2 i_3 i_4} I^{(D)}_{i_1 i_2 i_3 i_4} + \sum_{[i_1|i_3]} c_{i_1 i_2 i_3} I^{(D)}_{i_1 i_2 i_3} + \sum_{[i_1|i_2]} b_{i_1 i_2} I^{(D)}_{i_1 i_2} + \sum_{[i_1|i_1]} a_{i_1} I^{(D)}_{i_1} + \mathcal{R}_N$$

- master integrals known in literature
- and implemented in various codes, e.g. QCDLoop [Ellis, Zanderighi] (QCDLoop.fnal.gov)
- To do: determination of the master-integral coefficients
  - ⬅ generalized-unitarity techniques [Britto, Cachazo, Feng — Bern, Dixon, Dunbar, Kosower]
  - ⬅ subtraction terms to extract lower-point coefficients best identified at the integrand level [Ossola, Papadopoulos, Pittau]

note that $[i_1|i_n] = 1 \leq i_1 < i_2 < \ldots < i_n \leq N$ and $I^{(D)}_{i_1 \ldots i_n} = \int \frac{d^D\ell}{i\pi^{D/2}} \frac{1}{d_{i_1} \cdots d_{i_n}}$

Basics of Ellis–Giele–Kunszt–Melnikov (EGKM) method

- integrand is re-expressed by sum of basic denominator structures

$$\mathcal{A}_N^{(1)}(\{p_i\} \mid \ell) = \sum_{k=1}^{5} \sum_{[i_1|i_k]} \frac{\mathcal{P}(\vec{c}_{i_1\ldots i_k} \mid \ell)}{d_{i_1}\cdots d_{i_k}}$$

- numerators encode ℓ dependence ➡ parametric form: polynomial functions in coefficients

$$\mathcal{P}(\vec{c}_{i_1\ldots i_k} \mid \ell) \sim \sum_j \alpha_j(\ell) \times c^{(j)}_{i_1\ldots i_k} = \text{MI} + \text{rational} + \text{spurious} \quad \text{terms}$$

$\int d^D\ell \ \ldots$ MI terms $= c^{(0)}_{i_1\ldots i_k} I_{i_1\ldots i_k}$ and rational terms $= c^{(r)}_{i_1\ldots i_k}/\#$

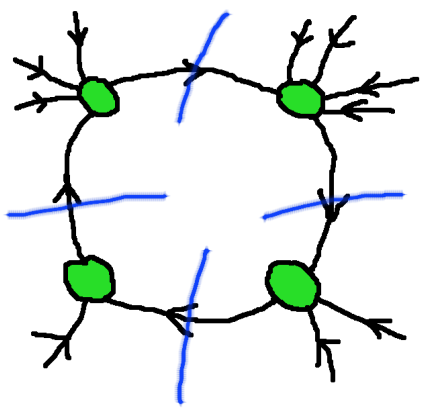
spurious terms vanish upon integration

- solve for coefficients by solving systems of equations given by $\ell = \tilde{\ell}$ such that $d_{i_1}, \ldots, d_{i_n} \equiv 0$

$$\mathcal{P}(\vec{c}_{i_1\ldots i_n} \mid \tilde{\ell}) = \sum_{\text{dof}}^{\text{internal}} \prod_{k=1}^{n} \mathcal{M}^{(0)}\left(\tilde{\ell}_{i_k}, \{p_j\}, -\tilde{\ell}_{i_{k+1}}\right)$$

Using tree-level MEs !

$$- \sum_{k=n+1}^{5} \sum_{[i_1|i_k]} d_{i_1}\cdots d_{i_n} \frac{\mathcal{P}(\vec{c}_{i_1\ldots i_k} \mid \tilde{\ell})}{d_{i_1}\cdots d_{i_k}}$$



➡ re-expressing the integrand

$$\mathcal{A}_N^{(1)(D_s)}(\{p_i\} \mid \ell) \;=\; \frac{\mathcal{N}_0(\{p_i\} \mid \ell) + (D_s - 4)\,\mathcal{N}_1(\{p_i\} \mid \ell)}{d_1 d_2 \ldots d_N} \;=$$

$$\sum_{[i_1|i_5]} \frac{\bar{e}^{(D_s)}_{i_1 i_2 i_3 i_4 i_5}(\ell)}{d_{i_1} d_{i_2} d_{i_3} d_{i_4} d_{i_5}} + \sum_{[i_1|i_4]} \frac{\bar{d}^{(D_s)}_{i_1 i_2 i_3 i_4}(\ell)}{d_{i_1} d_{i_2} d_{i_3} d_{i_4}} + \sum_{[i_1|i_3]} \frac{\bar{c}^{(D_s)}_{i_1 i_2 i_3}(\ell)}{d_{i_1} d_{i_2} d_{i_3}} + \sum_{[i_1|i_2]} \frac{\bar{b}^{(D_s)}_{i_1 i_2}(\ell)}{d_{i_1} d_{i_2}} + \sum_{[i_1|i_1]} \frac{\bar{a}^{(D_s)}_{i_1}(\ell)}{d_{i_1}}$$

- solving for numerator factors ➡ "the Left-Hand-Side"

$$\bar{e}^{(D_s)}_{i_1 \ldots i_5}(\ell) = \mathrm{Res}_{i_1 \ldots i_5}(\mathcal{A}_N^{(D_s)}(\ell))\,, \qquad \bar{d}^{(D_s)}_{i_1 \ldots i_4}(\ell) = \mathrm{Res}_{i_1 \ldots i_4}\left[\mathcal{A}_N^{(D_s)}(\ell) - \sum_{[j_1|j_5]} \frac{\bar{e}^{(D_s)}_{j_1 j_2 j_3 j_4 j_5}(\ell)}{d_{j_1} d_{j_2} d_{j_3} d_{j_4} d_{j_5}}\right], \quad \ldots$$

need to find $D \leq D_s$ dim. $\ell = \tilde{\ell} = \ell_{i_1 \ldots i_n}$ such that $d_j(\tilde{\ell}) \equiv 0$ for $j = i_1, \ldots, i_n$

define $\mathrm{Res}_{i_1 \ldots i_n}\left(\mathcal{A}_N^{(D_s)}(\ell)\right) = \left\{d_{i_1}(\ell) \cdots d_{i_n}(\ell) \times \mathcal{A}_N^{(D_s)}(\ell)\right\}\Big|_{\ell = \tilde{\ell} = \ell_{i_1 \ldots i_n}}$

- find parametric form of residues, removing spurious terms ➡ "the Right-Hand-Side"

box coefficient: $\bar{d}^{(D_s)}_{i_1 \ldots i_4}(\ell) \;=\; d^{(0)}_{i_1 \ldots i_4} + \alpha_4\, d^{(1)}_{i_1 \ldots i_4} + s_e^2\,[d^{(2)}_{i_1 \ldots i_4} + \alpha_4\, d^{(3)}_{i_1 \ldots i_4}] + s_e^4\, d^{(4)}_{i_1 \ldots i_4}$

$$\Rightarrow \quad \int d^D\ell \frac{\bar{d}^{(D_s)}_{i_1 \ldots i_4}(\ell)}{d_{i_1} d_{i_2} d_{i_3} d_{i_4}} \;=\; d^{(0)}_{i_1 \ldots i_4}\, I_{i_1 \ldots i_4} \;-\; d^{(4)}_{i_1 \ldots i_4}/6$$

Calculation of the residues

- What is $\mathrm{Res}_{i_1\ldots i_n}\left(\mathcal{A}_N^{(D_s)}(\ell)\right)$?

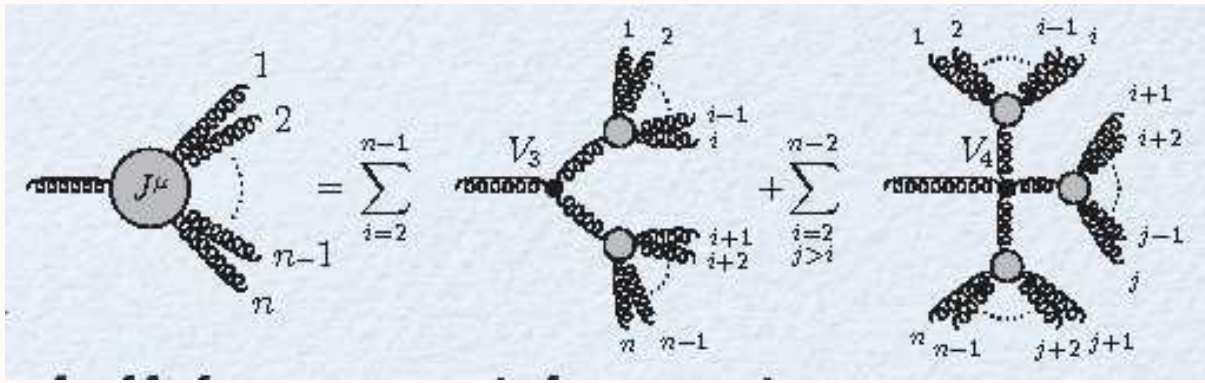
$$= \left\{ d_{i_1}(\ell)\cdots d_{i_n}(\ell) \times \mathcal{A}_N^{(D_s)}(\ell) \right\}\Big|_{d_{i_1}(\ell)=\cdots=d_{i_n}(\ell)=0}$$

- requires calculation of factorized un-integrated one-loop amplitude
- unitarity cuts: M on-shell propagators, amplitude factorizes into M tree-level amplitudes

$$\mathrm{Res}_{i_1\ldots i_M}\left(\mathcal{A}_N^{(D_s)}(\ell)\right) = \sum_{\{\lambda_1,\ldots,\lambda_M\}=1}^{D_s-2} \left(\prod_{k=1}^{M} \mathcal{M}^{(0)}\left(\ell_{i_k}^{(\lambda_k)}; p_{i_k+1},\ldots,p_{i_{k+1}}; -\ell_{i_{k+1}}^{(\lambda_{k+1})}\right)\right)$$

- two D_s dimensional gluons with complex momenta and D_s-2 polarization states $\quad (\ell_{i_k} = \ell + \tilde{q}_{i_k})$

- Berends–Giele recursion relations to calculate tree-level amplitudes
- very economical scheme
- LHS:
  take subtractions into account



Algorithm for full one-loop amplitudes

➡ ***EGKM implementations to calculate ordered amplitudes are robust and sufficiently fast***

using Berends–Giele recursion relations to determine the $\mathcal{M}^{(0)}$ pieces yields algorithm of polynomial complexity ($\tau \sim N^{\#}$) [GIELE, ZANDERIGHI — LAZOPOULOS — GIELE, WINTER]

➡ In general, the sum over colour orderings has to be performed in some way $\Rightarrow$ obtain $\mathbf{2\,Re(\mathcal{M}^{(0)^*}\mathcal{M}^{(1)})}$ ... ***may become laborious*** ... all orderings need be known

- (naive) permutation sum re-introduces factorial growth ... $(N-1)!/2$
- complexity of colour decomposition increases for quark dominated processes

- Can we do better ... tame the growth ?

Construction of an algorithm of exponential complexity, colour quantum #s included.

$\Rightarrow$ Naive expectation of the asymptotic scaling is $(f \times 5)^N$ for N legs.

$\Rightarrow$ Colour-dressed recursions give factor $f > 1$, can be as large as 4.

$\Rightarrow$ Number of pentagons rise with 5^N ... asymptotic behaviour of Stirling # $\mathcal{S}_2(N,5)$.

input: external parton momenta & polarizations plus their explicit colours (colour-flow representation)

output: amplitude $\mathcal{M}^{(1)}$ in the form of a complex number (FDH scheme)

EGKM extended

[GIELE, KUNSZT, WINTER, ARXIV:0911.1962]

➡ ***Start off the EGKM algorithm for colour-ordered amplitudes.***
To include full colour information, extensions are necessary:

- Decomposition of the integrand: sums over ordered cuts change into sums over partitions including non-cyclic, non-reflective permutations of the initial partitions.

$$\sum_{[i_1|i_k]} \quad \rightarrow \quad \sum_{RP_{\pi_1\ldots\pi_k}(1,2,\ldots,N)}$$

- Identification of the subtraction terms when solving for $\mathcal{P}(\vec{c}_{\pi_1\ldots\pi_k} \mid \tilde{\ell})$: identify by de-pinching, account for possible shifts in loop momenta.

  e.g. 4-gluon bubble 01|23 has 4 triangle subtraction terms:
  0|1|23 with $\hat{\ell} = \ell$ and $\hat{\ell} = -\ell + p_{23}$ and 2|3|01 with $\hat{\ell} = -\ell$ and $\hat{\ell} = \ell + p_{01}$

- Calculation of the integrand's residues: use colour-dressed recursions and sum over internal polarizations and internal colours.

$$\sum_{\text{dof}}^{\text{internal}} \prod_{k=1}^{n} \mathcal{M}^{(0)}\left(\tilde{\ell}_{i_k}, \{p_j\}, -\tilde{\ell}_{i_{k+1}}\right) \quad \rightarrow \quad \sum_{\substack{\{\lambda_j\} \\ \{(IJ)_j\}}} \prod_{k=1}^{n} \mathcal{M}^{(0)}\left(\tilde{\ell}_{\pi_k}^{(\lambda_k (IJ)_k)}, p_{\pi_k}, -\tilde{\ell}_{\pi_{k+1}}^{(\lambda_{k+1} (JI)_{k+1})}\right)$$

- Decomposition of one-loop amplitude: comes with symmetry factor of $1/2!$ in front of the bubble-coefficient terms.

Unordered gluons: a note on partitions

- number of unitarity cuts, example 4-gluon loop

  ordered) 0|1|2|3
  01|2|3, 0|12|3, 0|1|23, 1|2|30
  ~~0|123~~, ~~1|230~~, ~~2|301~~, ~~3|012~~, 01|23, 12|30

  unordered) 0|1|2|3, 0|2|3|1, 0|3|1|2
  0|1|23, 0|2|13, 0|3|12, 1|2|03, 1|3|02, 2|3|01
  ~~0|123~~, ~~1|023~~, ~~2|013~~, ~~3|012~~, 01|23, 02|13, 03|12

ord.)	N	5-gons	boxes	triangles	bubbles	total
3	4	0	1	4	2	7
12	5	1	5	10	5	21
60	6	6	15	20	9	50
360	7	21	35	35	14	105
2520	8	56	70	56	20	202
20160	9	126	126	84	27	363

unord.)	N	5-gons	boxes	triangles	bubbles	total
	4	0	3	6	3	12
	5	12	30	25	10	77
	6	180	195	90	25	490
	7	1680	1050	301	56	3087
	8	12600	5103	966	119	18788
	9	83412	23310	3025	246	109993

ord.) number of orderings however grows as $(N-1)!/2$, unord.) Stirling numbers grow as k^N

- number of k-cut combinations: $\mathcal{C}(N,k) = \binom{N}{k} - N\,\Theta(2-k)$
  $\Rightarrow$ but to multiply with number of orderings
- number of $k \geq 2$-cut partitions: $\max\{1,(k-1)!/2\} \times \mathcal{S}_2(N,k) - N\,\Theta(2-k)$
  $\Rightarrow$ increased number of terms, origin of exponential growth

Colour-dressed recursion relations

- show exponential growth with N, cf. [Duhr, Höche, Maltoni], implemented in ...

➡ ***COMIX ... SM tree-level ME generator based on generalized colour-dressed Berends–Giele recursions*** [Gleisberg, Höche]

- colour-flow decomposition for gluon currents used in our study

$$J_\mu^{IJ}(1,2,..,n) = \sum_{\sigma\in S_n} \delta^I_{j_{\sigma_1}} \delta^{i_{\sigma_1}}_{j_{\sigma_2}} \cdots \delta^{i_{\sigma_{n-1}}}_{j_{\sigma_n}} \delta^{i_{\sigma_n}}_J \; J_\mu(\sigma_1,\sigma_2,..,\sigma_n)$$

$$= \kappa^{-2}(1,2,..,n)\Big[\sum_{P_{\pi_1\pi_2}} \left(\delta^I_K\delta^L_M\delta^N_J - \delta^I_M\delta^N_K\delta^L_J\right)\left[J_\mu^{KL}(\pi_1), J_\mu^{MN}(\pi_2)\right] +$$

$$\sum_{P_{\pi_1\pi_2\pi_3}} \left(\delta^{ILNP}_{KMOJ} + \delta^{IPNL}_{OMKJ} - \delta^{ILPN}_{KOMJ} - \delta^{INPL}_{MOKJ}\right)\left(\left\{J_\mu^{KL}(\pi_1), J_\mu^{MN}(\pi_2), J_\mu^{OP}(\pi_3)\right\} + \pi_1 \leftrightarrow \pi_2\right)\Big]$$

- our tree-level amplitude calculations scale as 4^N
  (in COMIX, V_{gggg} is replaced by effective V_{ggg}, which yields 3^N scaling)
- used to calculate the LHS of the parametric form when solving for the coefficients

$$\mathrm{Res}_{\kappa_1\cdots\kappa_n}\left(\mathcal{A}_N^{(D_s)}(\ell)\right) = \sum_{\substack{\{\lambda_j=1\}\\ \{(IJ)_j\}}}^{D_s-2} \prod_{i=1}^{n} \mathcal{M}^{(0)}\left(\tilde{\ell}_{\pi_i}^{(\lambda_i (IJ)_i)}, p_{\pi_i}, -\tilde{\ell}_{\pi_{i+1}}^{(\lambda_{i+1} (JI)_{i+1})}\right)$$

- internal colour sum is costly: reuse as many J_μ^{IJ} as possible, store & compute only non-zeros

C++ code

Implementation of ordered algorithm based on ...

[Ellis, Giele, Kunszt, arXiv:0708.2398] 4dim method, cut-constructible part

[Giele, Kunszt, Melnikov, arXiv:0801.2237] Ddim method, rational part

[Giele, Zanderighi, arXiv:0805.2152] application of Ddim method to pure gluons

- independent implementation and cross check of EGKM method
  (from scratch, no translation of Fortran routines)
- documented in [Giele, Winter, arXiv:0902.0094] plus discussion of reasons for precision loss for larger N

Colour-dressed algorithm for N external gluons ...

- stringent test — colour-dressed and colour-decomposition results have to agree

  (1) $\Rightarrow$ all orders of ϵ, schematically $\mathcal{M}^{(1)} = \sum_{P(2,..,N)/Z_{N-1}} \left\{ \sum_r^{2^N} N_{\rm C}^{b(r)} \prod_s^N \delta_{j_s(r)}^{i_s(r)} \right\} \mathcal{A}^{(1)}(1,\ldots,N)$

  (2) $\Rightarrow$ double poles obey $\mathcal{M}_{\rm dp}^{(1)} = -c_\Gamma\, \epsilon^{-2}\, N_{\rm C} N\, \mathcal{M}^{(0)}$

- efficiency – scaling of computing time with # of legs N $\rightarrow$ $\tau \sim x^N$
- accuracy – numerical stability of algorithm
- phase-space integration tests using colour sampling

Scaling behaviour of the algorithm

- Table taken from an early test: $2 \to N-2$ gluons

  $(++--..)$ polarizations, $\binom{..1131..}{..1311..}$ colours & random PSPs obeying separation cuts ...

  computation times in secs (2.20 GHz Intel Core2 Duo)

ord.)	N	cut-c,4D	factor	full,5D	factor	unord.)	N	cut-c,4D	factor	full,5D	factor	OK?
2	4	0.025		0.045			4	0.05		0.105		✓
6	5	0.185	7.4	0.355	7.9		5	0.315	6.3	0.74	7.0	✓
24	6	0.83	4.5	2.7	7.6		6	1.37	4.3	4.59	6.2	✓
120	7	7.95	9.6	27.5	10.2		7	8.4	6.1	32.5	7.1	✓
720	8	86.5	10.9	328	11.9		8	52	6.2	234	7.2	✓
5040	9	1070	12.4	4250	13.0		9	354	6.8	1720	7.4	✓
40320	10	14000	13.1	60600	14.3		10			13700	8.0	✓

ord.) factors clearly increase with larger N, unord.) growth follows $(f \cdot 5)^N$, $1 < f < 2$

number of non-zero colour factors grows as $(N-2)!$ for this case

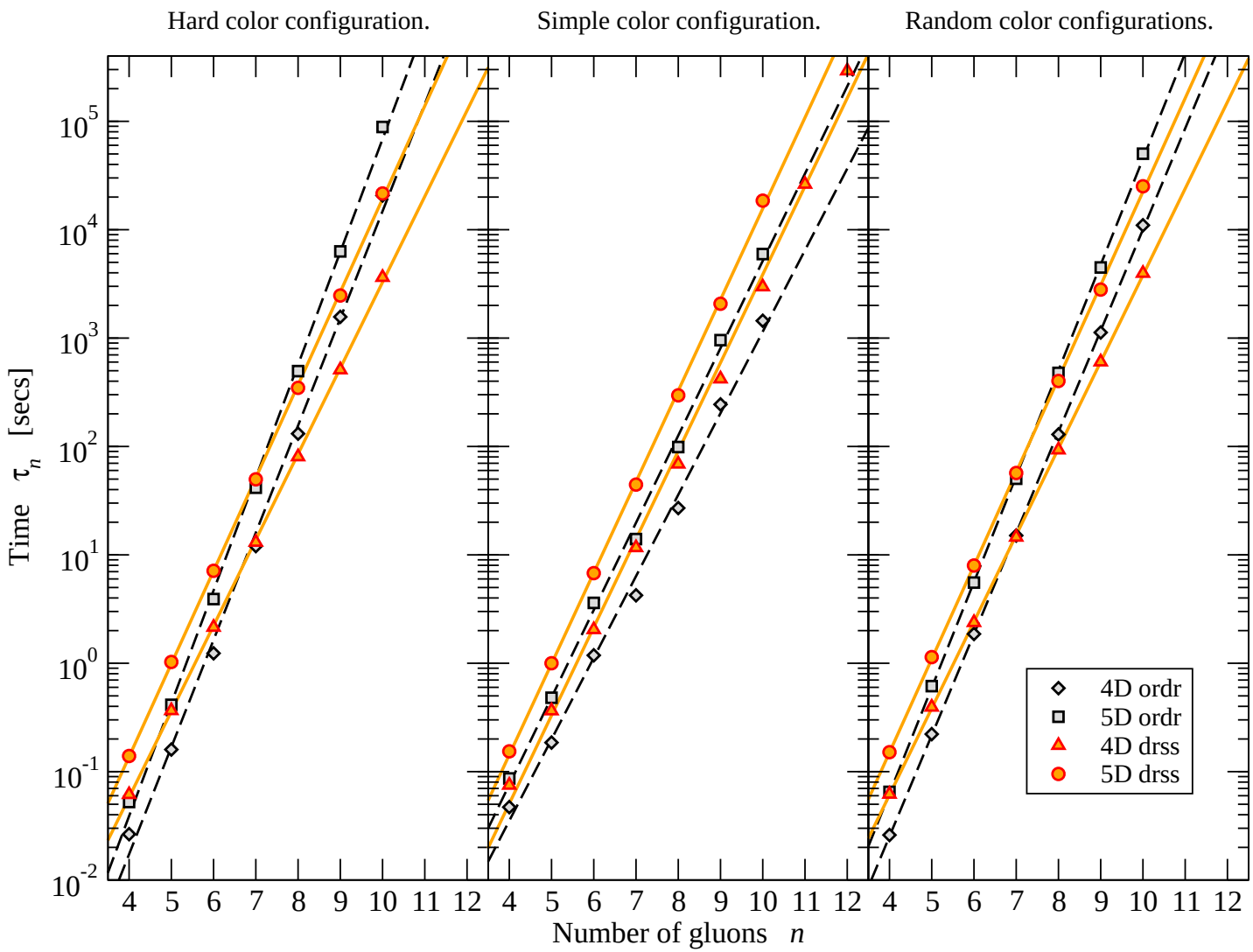
Scaling of the computation time with # of legs

(calculations in double precision) [GIELE, KUNSZT, WINTER, ARXIV:0911.1962]

- algorithm checked for exponential complexity ($\tau \sim x^N$)

- Random colours:

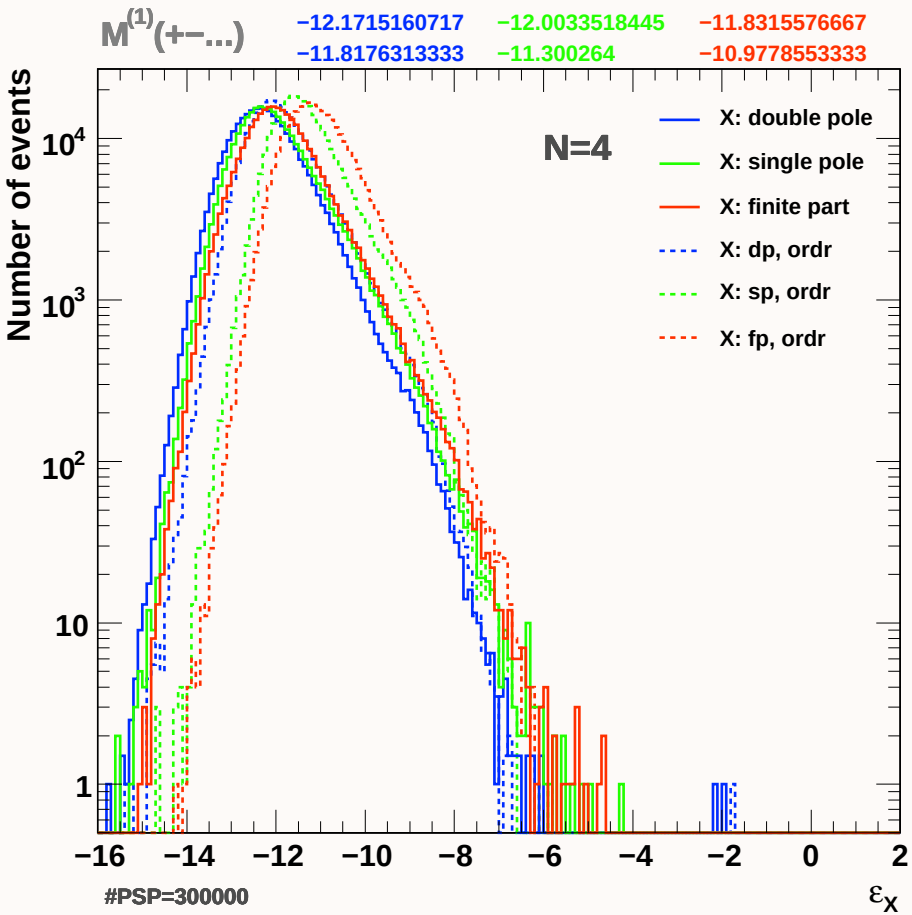
Algorithm	x
4dim ordered	8.6(1)
5dim ordered	9.5(1)
4dim unord.	6.30(4)
5dim unord.	7.3(1)

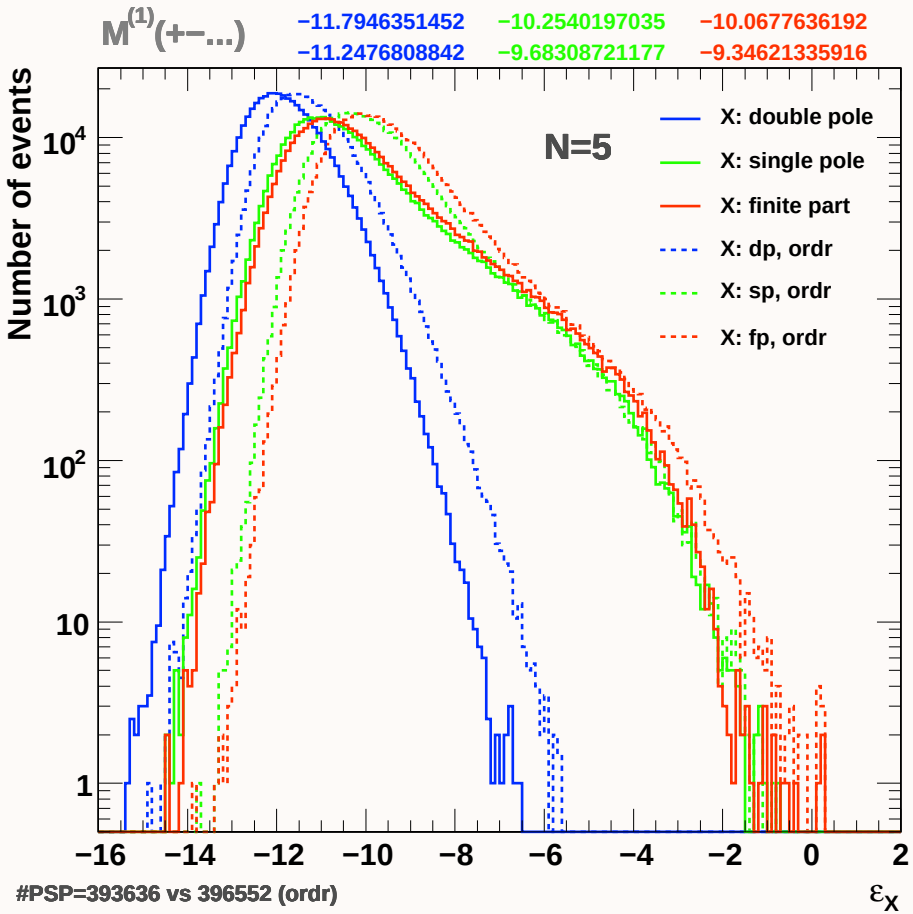


Accuracy

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- unordered algorithm provides on average more accurate results
- peak positions & tails are OK,

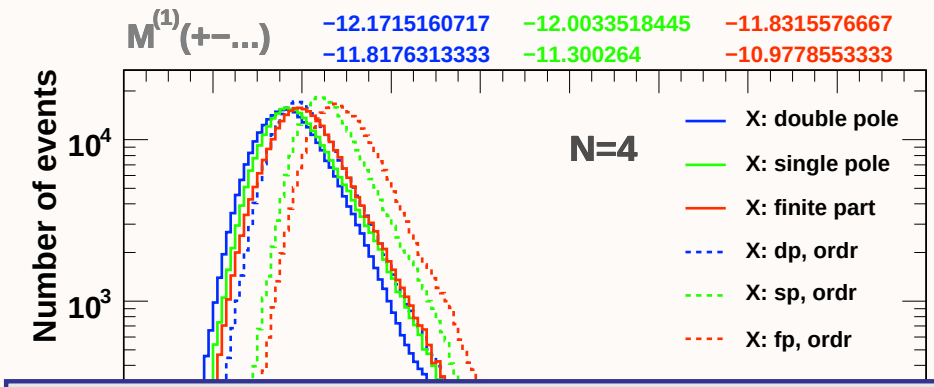


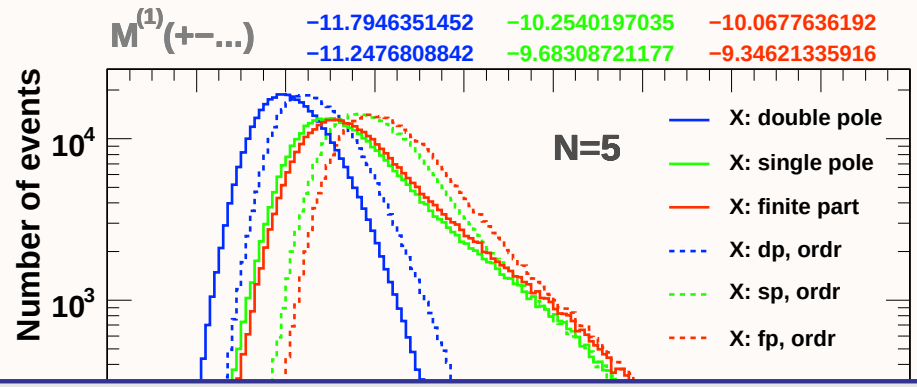


Accuracy

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- unordered algorithm provides on average more accurate results
- peak positions & tails are OK,





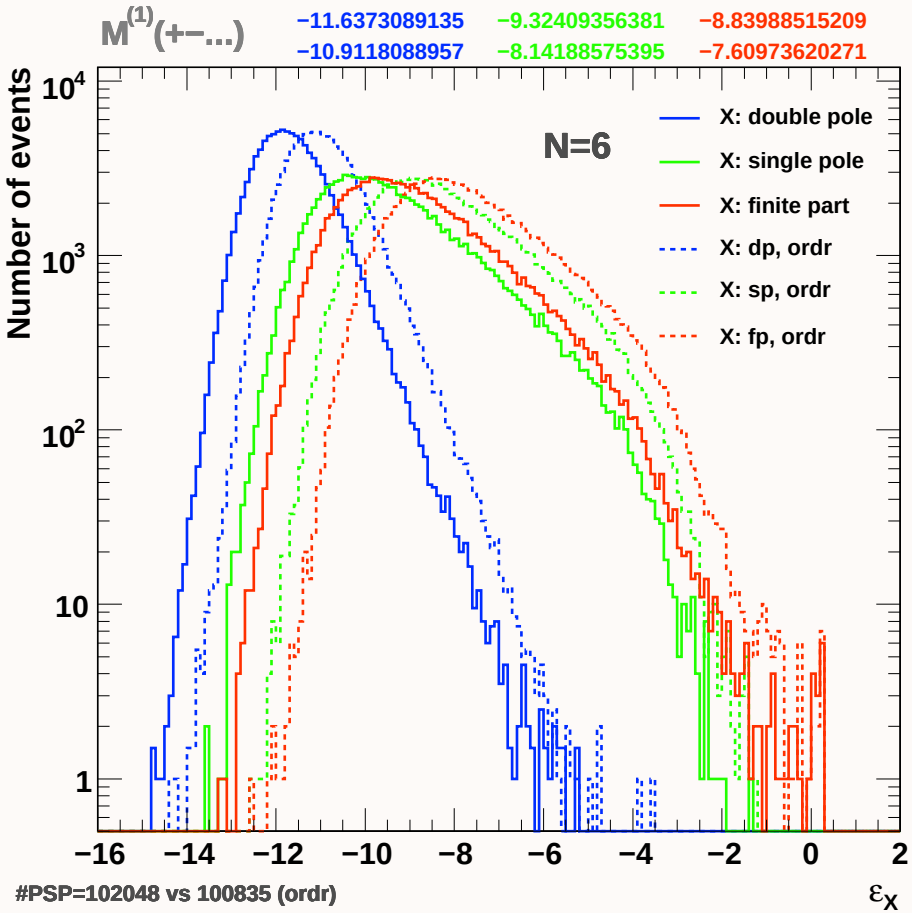
- accuracy — numerical stability of algorithm

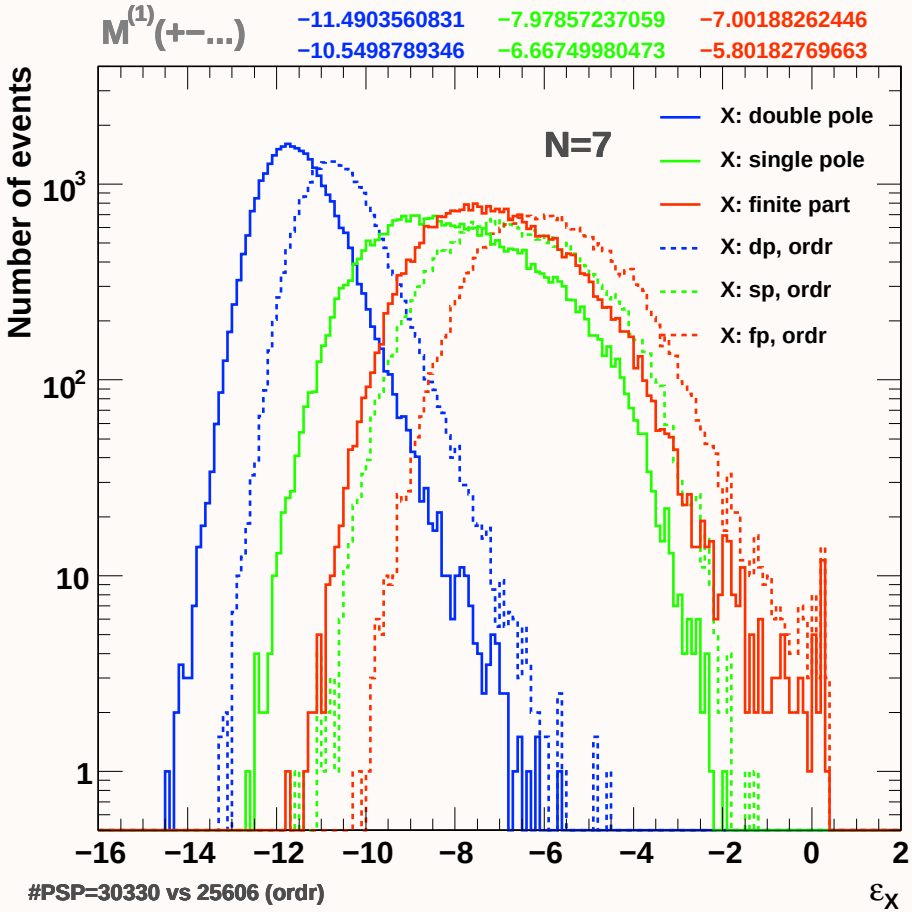
$$\varepsilon_{\rm dp} = \log_{10} \frac{|\mathcal{M}^{(1)[1]}_{\rm dp,num} - \mathcal{M}^{(1)}_{\rm dp,th}|}{|\mathcal{M}^{(1)}_{\rm dp,th}|}, \qquad \varepsilon_{\rm s/fp} = \log_{10} \frac{2\,|\mathcal{M}^{(1)[1]}_{\rm s/fp,num} - \mathcal{M}^{(1)[2]}_{\rm s/fp,num}|}{|\mathcal{M}^{(1)[1]}_{\rm s/fp,num}| + |\mathcal{M}^{(1)[2]}_{\rm s/fp,num}|}$$

Accuracy

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- unordered algorithm provides on average more accurate results
- peak positions & tails are OK, 97% ($N = 6$) and 89% (unord.) vs. 96% and 87% (ord.) of events can be handled in double precision

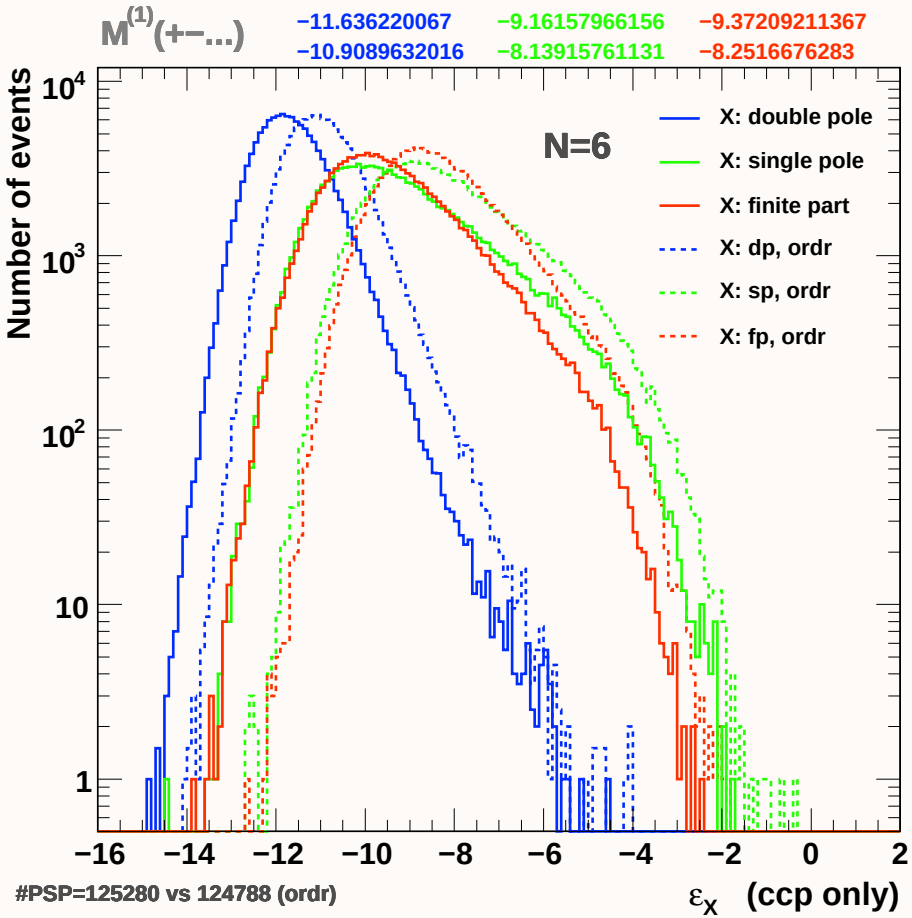


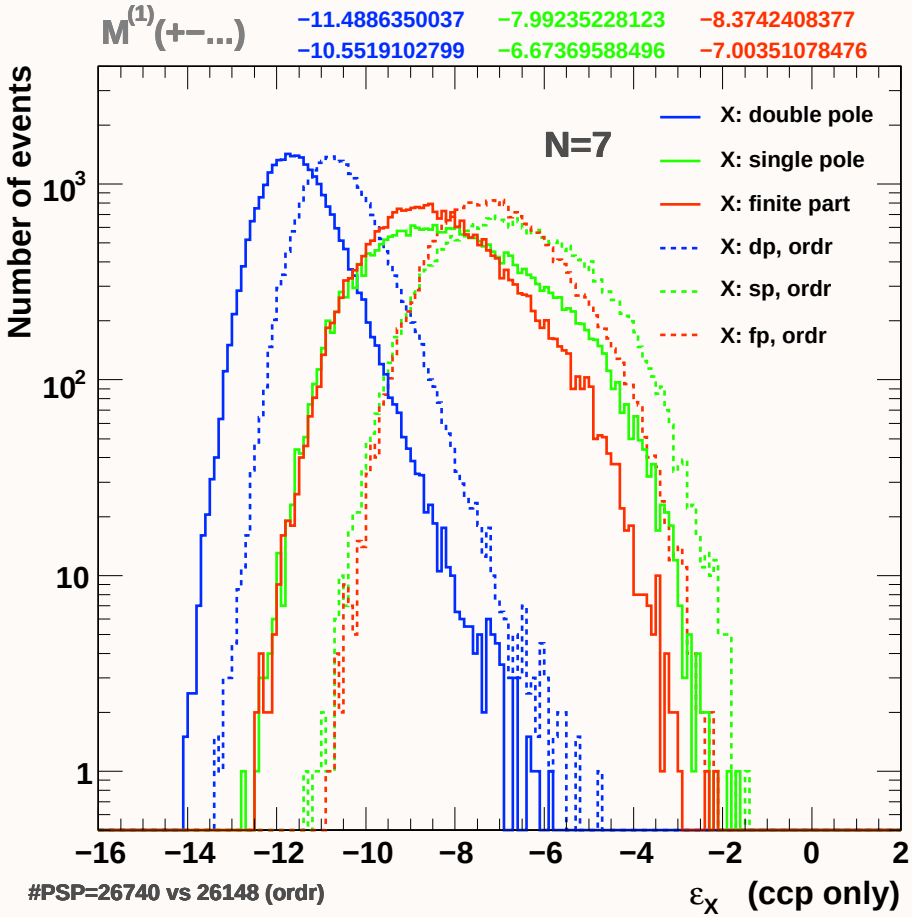


Accuracy

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- unordered algorithm provides on average more accurate results
- peak positions & tails are OK, 97% ($N = 6$) and 89% (unord.) vs. 96% and 87% (ord.) of events can be handled in double precision

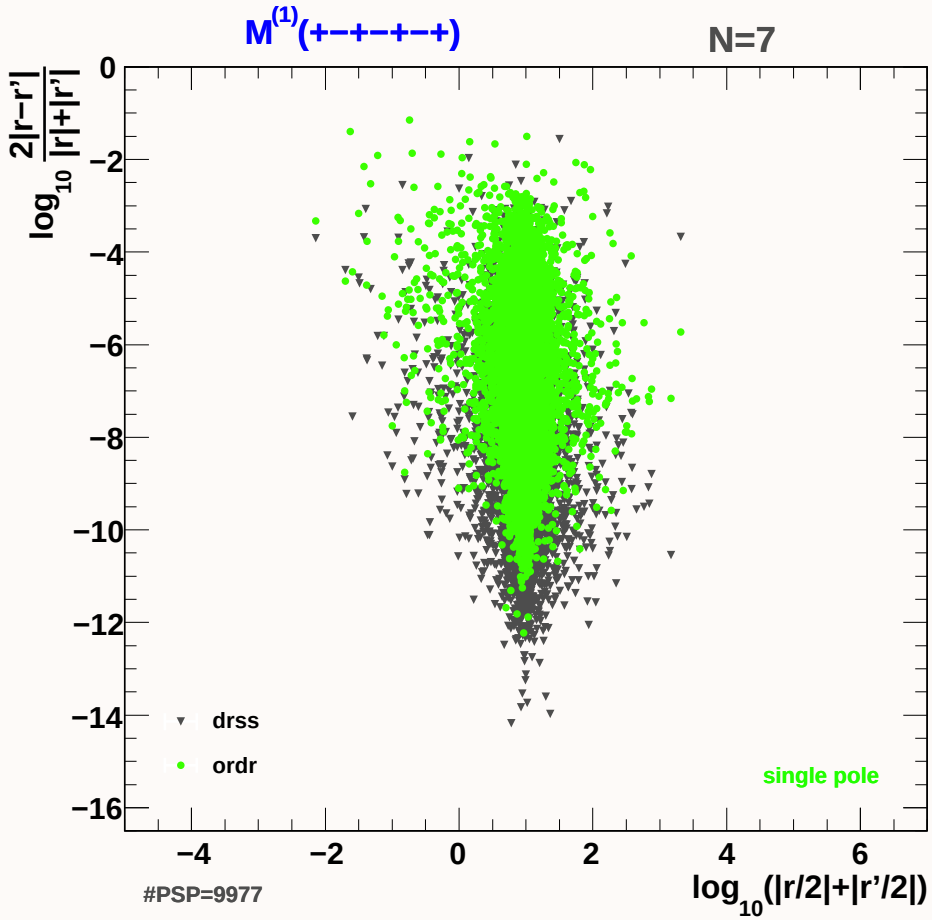


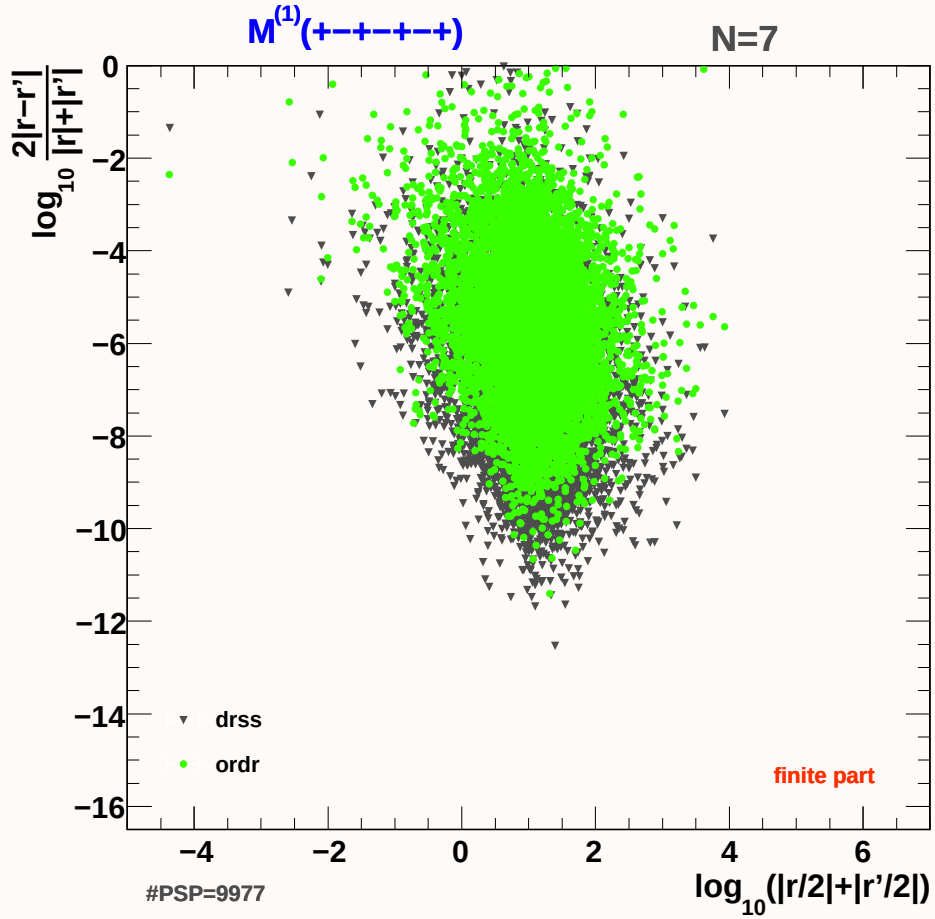


Accuracy

(calculations in double precision) [GIELE, KUNSZT, WINTER, ARXIV:0911.1962]

- accuracy of determining corrections versus their magnitude (accuracy vs. weight)
  $r = \mathrm{Re}\left(\mathcal{M}^{(0)\dagger}\mathcal{M}^{(1)}\right)/\left(2\pi|\mathcal{M}^{(0)}|^2\right)$
- unordered algorithm provides on average more accurate results

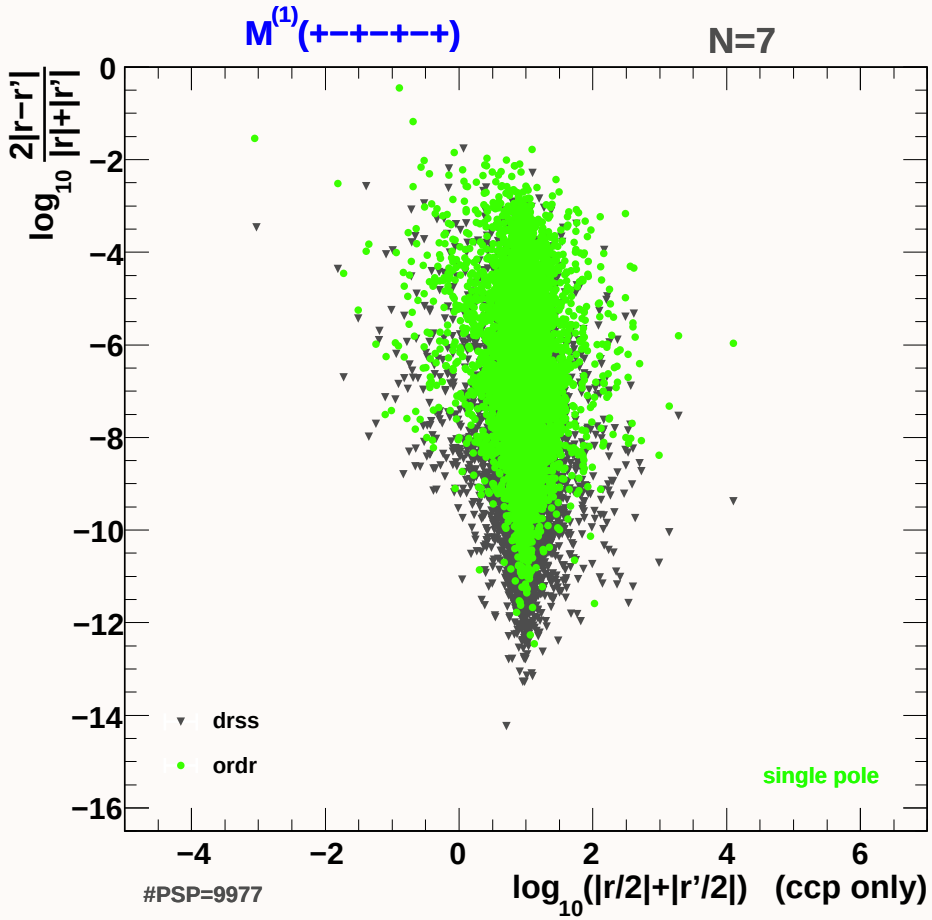


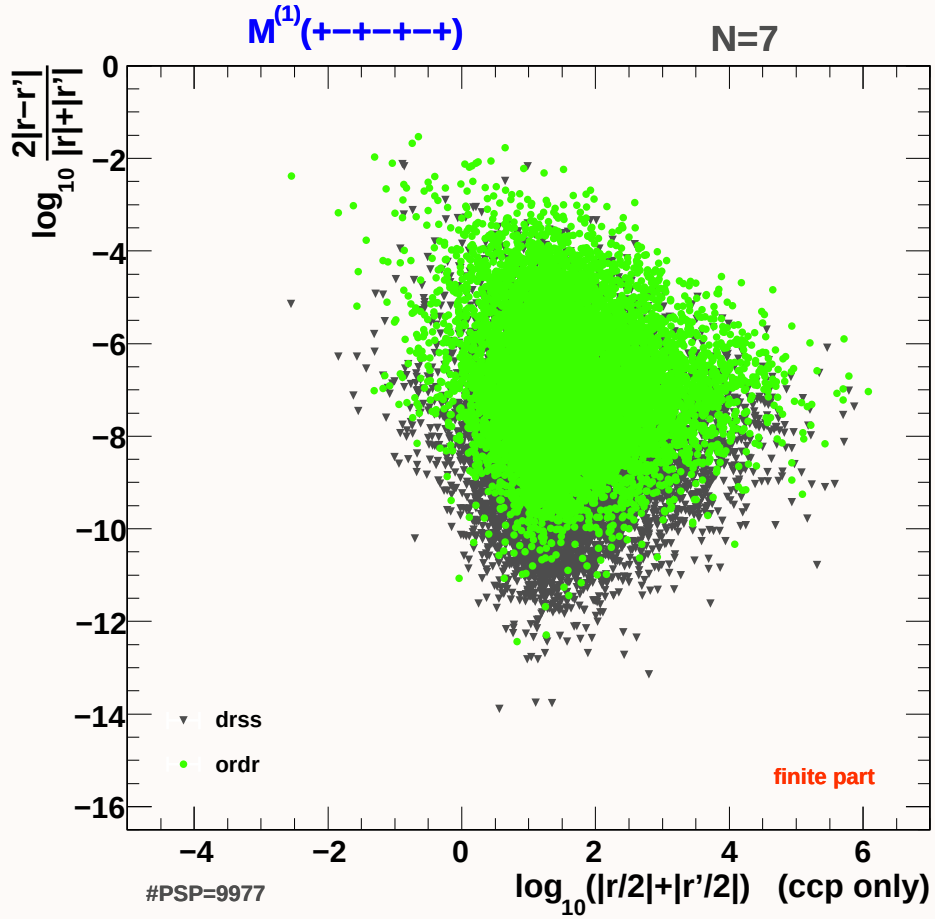


Accuracy

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- accuracy of determining corrections versus their magnitude (accuracy vs. weight)
  $r = \mathrm{Re}\big(\mathcal{M}^{(0)\dagger}\mathcal{M}^{(1)}\big)/\big(2\pi|\mathcal{M}^{(0)}|^2\big)$
- unordered algorithm provides on average more accurate results

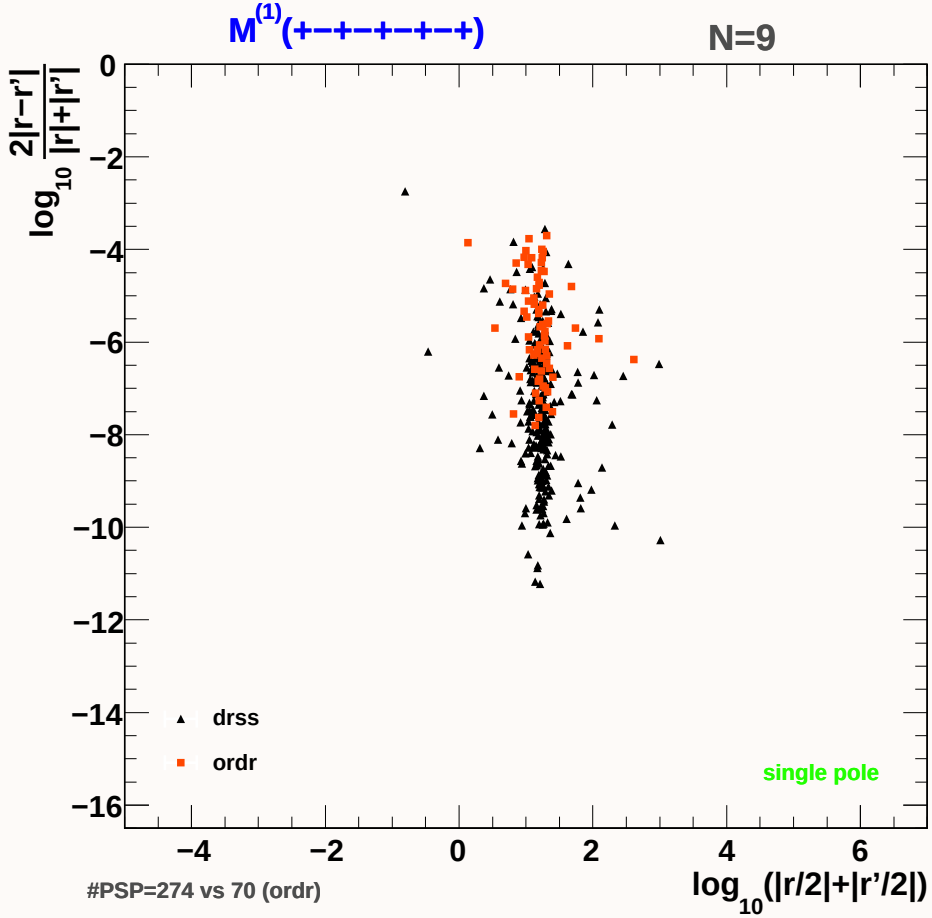


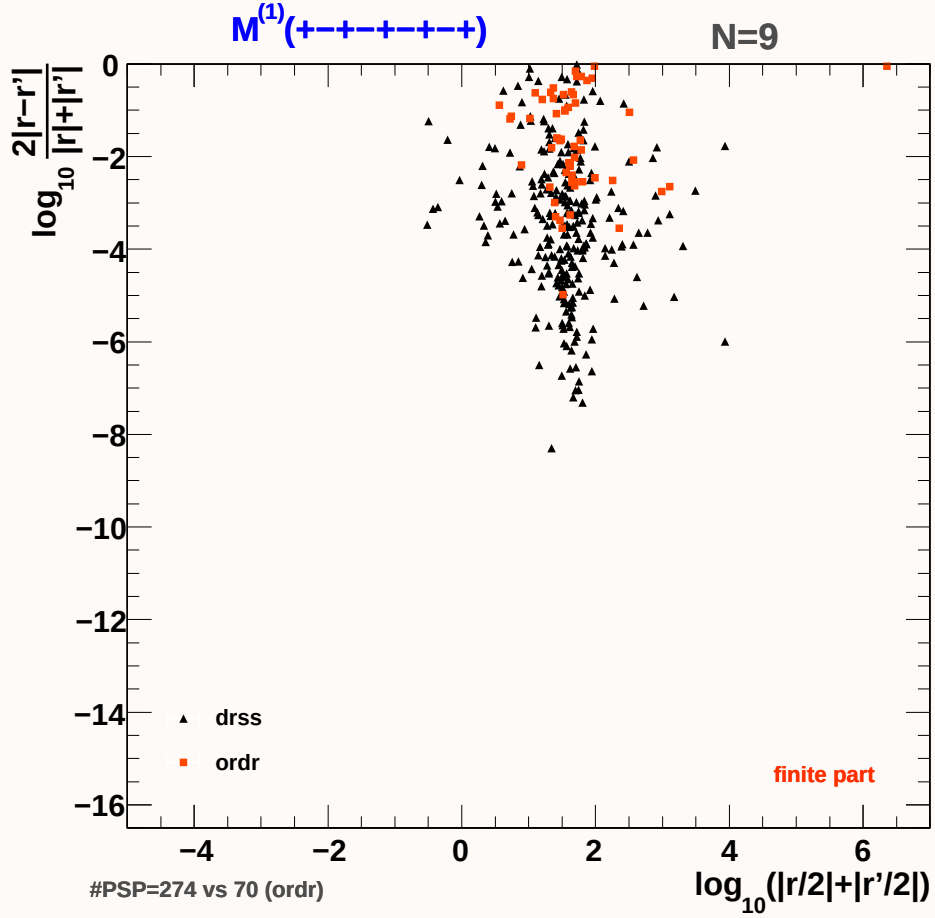


Accuracy

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- accuracy of determining corrections versus their magnitude (accuracy vs. weight)
  $r = \mathrm{Re}\big(\mathcal{M}^{(0)\dagger}\mathcal{M}^{(1)}\big)/\big(2\pi|\mathcal{M}^{(0)}|^2\big)$
- unordered algorithm provides on average more accurate results





Phase-space integration and colour sampling tests

(calculations in double precision) [Giele, Kunszt, Winter, arXiv:0911.1962]

- stability & consistency check: test convergence of uniform phase-space Monte Carlo integrations
- colour sampled:

  $$S_{\mathrm{MC}} = W_{\mathrm{col}} \times \mathcal{K}$$

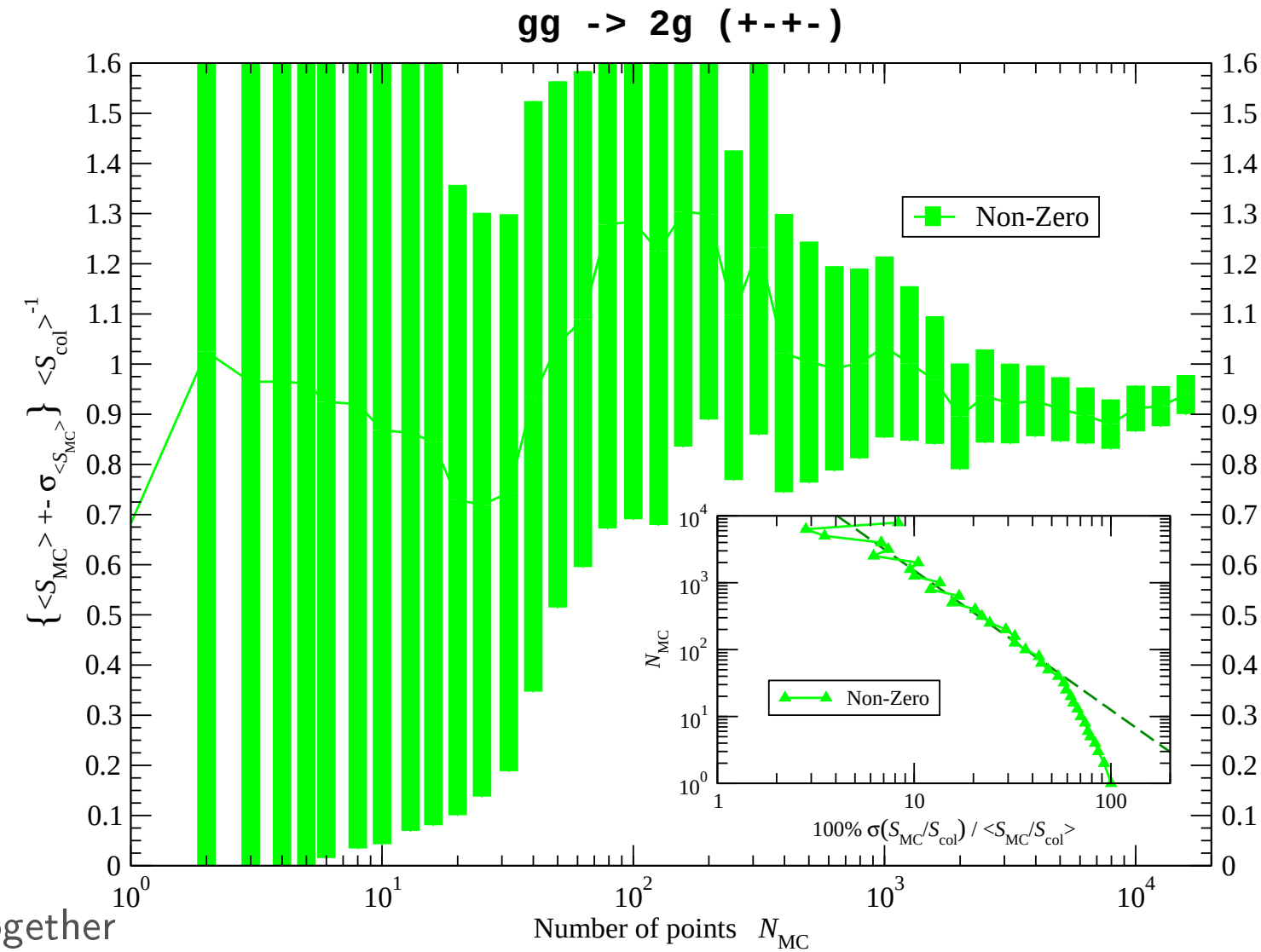
  normalized to colour summed:

  $$S_{\mathrm{col}} = \sum_{\mathrm{col}} \mathcal{K}$$

  with the kernel

  $$\mathcal{K} = |\mathcal{M}^{(0)}| + \frac{\alpha_s}{2\pi} \times \mathrm{Re}(\mathcal{M}^{(1)}_{\mathrm{fp}} \mathcal{M}^{(0)\dagger})$$

- only display standard deviation of $\langle S_{\mathrm{MC}} \rangle$
  ➡ MC phase-space integration and colour sampling work together



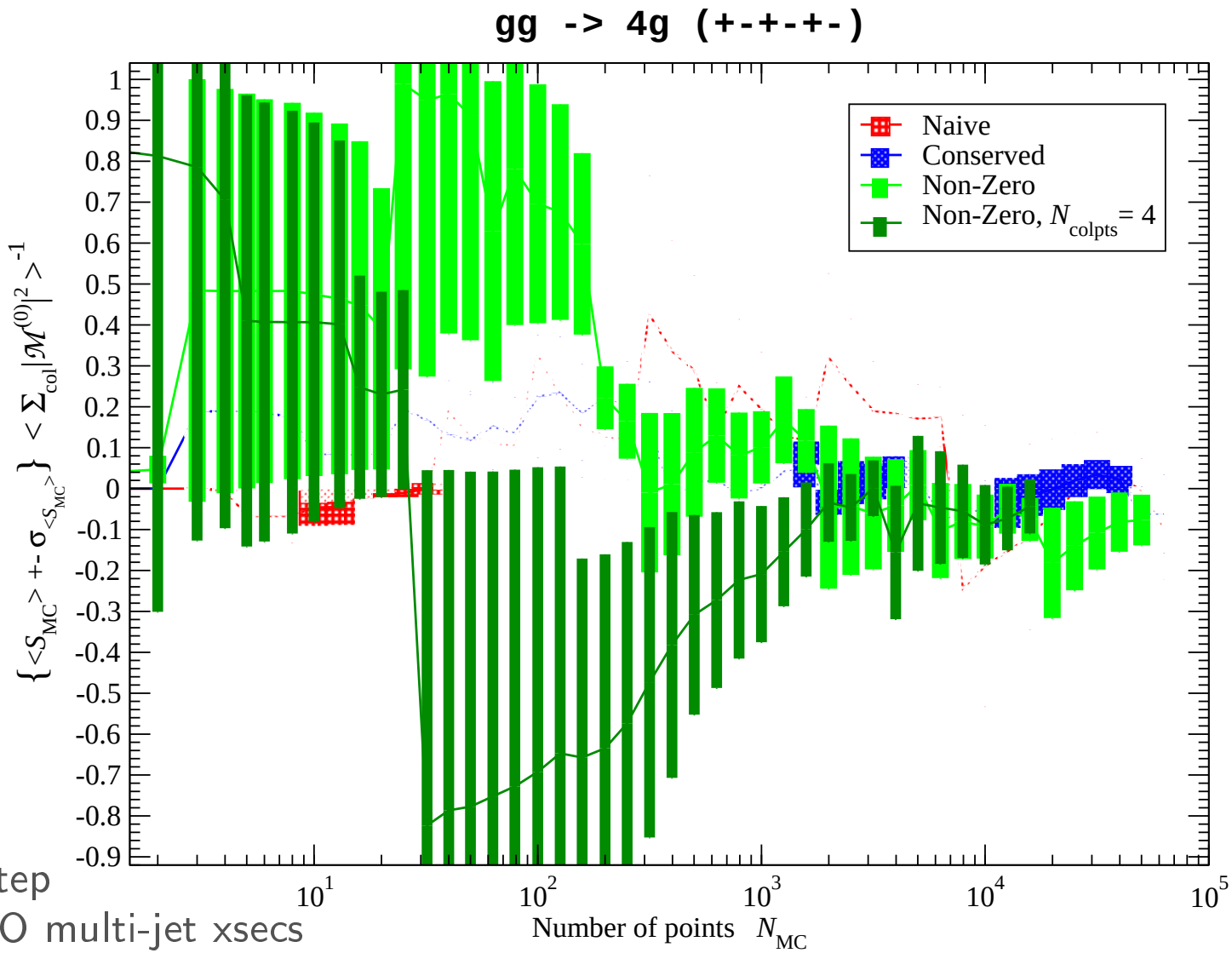
Phase-space integration and colour sampling tests

(calculations in double precision) [GIELE, KUNSZT, WINTER, ARXIV:0911.1962]

- stability check: test convergence of virtual corrections when integrated over a flat phase space
- colour sampled:

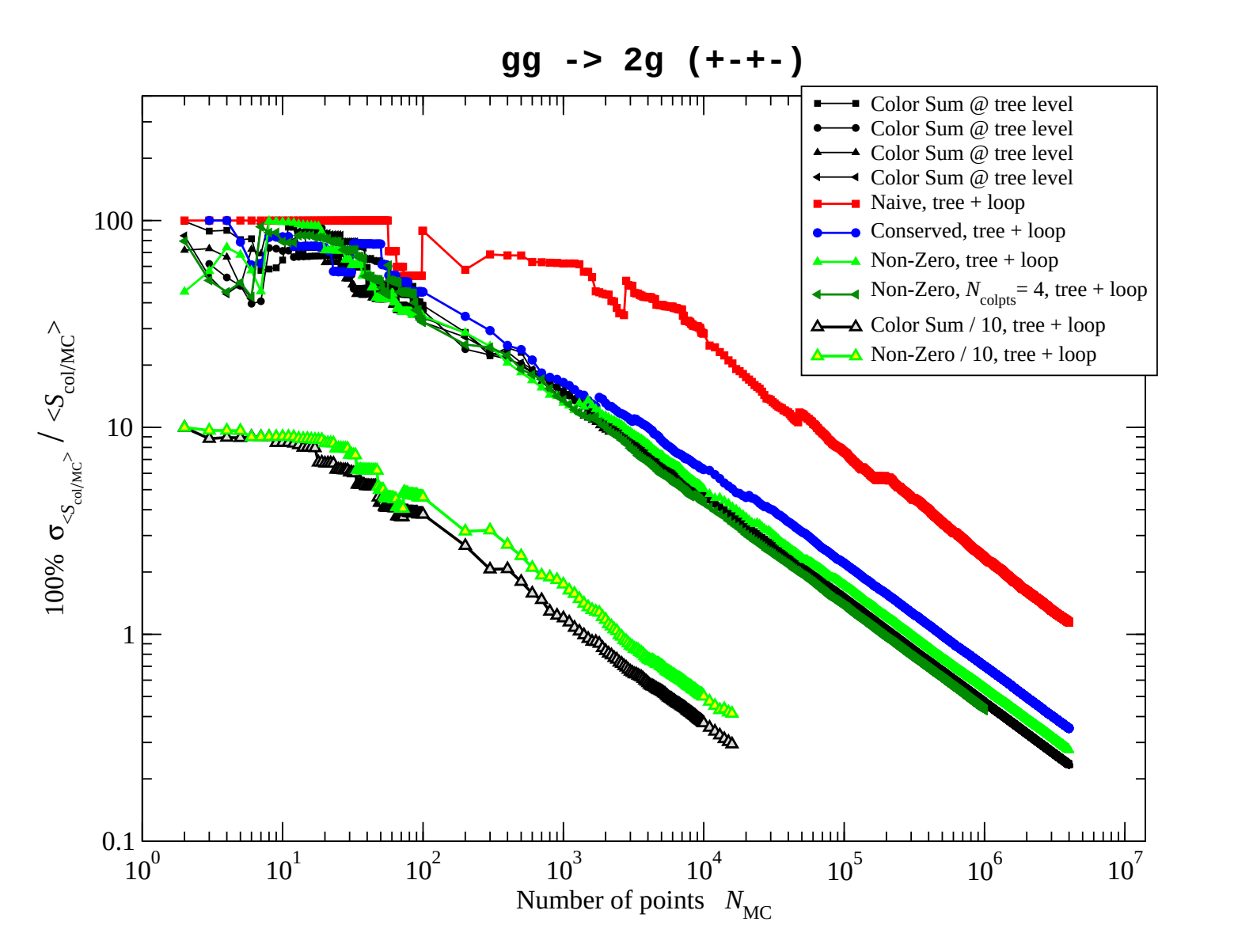
  $S_{\mathrm{MC}} = W_{\mathrm{col}} \times \mathcal{K}$

  normalized to colour summed Born contribution
- good estimate of magnitude of virtual correction
- different sampling schemes ➡ W_{col}
- only display standard deviation of $\langle S_{\mathrm{MC}} \rangle$
- first test of one major step in the calculation of NLO multi-jet xsecs



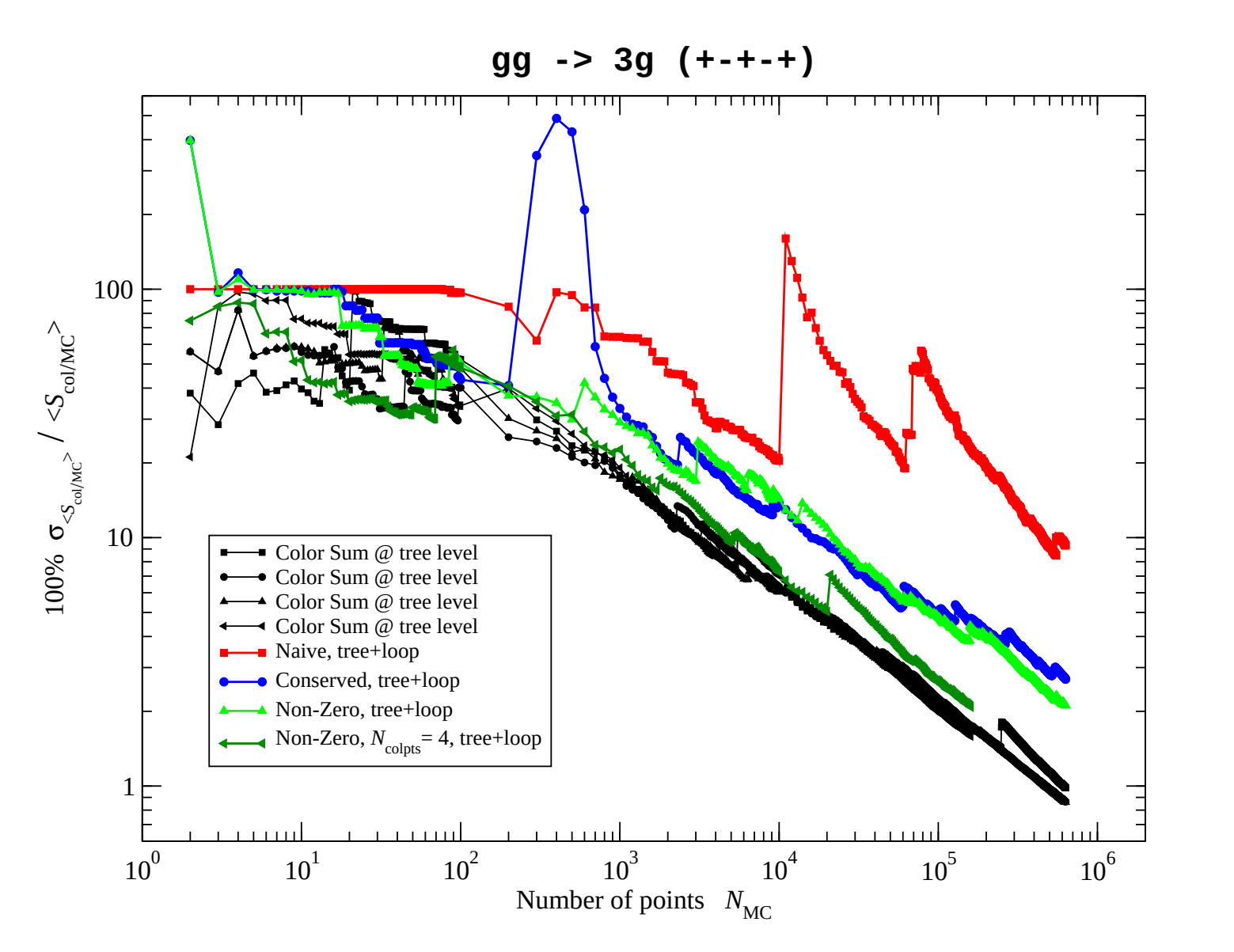
Relative errors

[GIELE, KUNSZT, WINTER, PRELIMINARY]



Relative errors

[GIELE, KUNSZT, WINTER, PRELIMINARY]



Summary

- Higher-order calculations are needed to meet the requirements on the precision of theoretical predictions in the LHC era.

- Highly automated and optimized parton-level event generators are available at tree level. At one loop, similar achievements seem possible owing to the new methods based on generalized unitarity and parametric integration techniques that use tree-level amplitudes as their input.

- Calculations based on recursive methods are easier to automate.
  Presented recursive scheme for the computation of QCD one-loop amplitudes that incorporates colour along with all other degrees of freedom.
  - $\Rightarrow$ algorithm is an extension of the Ellis–Giele–Kunszt–Melnikov method.

- Algorithmic implementation for full amplitudes using colour-dressed recursion relations.
  - $\Rightarrow$ algorithm is of exponential complexity.
  - $\Rightarrow$ asymptotic scaling of $\sim 7^N..8^N$ seen — milder than for colour decomposition.
  - $\Rightarrow$ more to do: fully include quarks, squared amplitudes, OLE, xsecs (pure jets).
  - $\Rightarrow$ potential improvements: fitting coefficients, higher precision.

- Numerical results presented for colour-dressed one-loop gluon amplitudes. Algorithm works.
  - $\Rightarrow$ reasonably accurate double-precision results — more accurate than for colour decomposition.
  - $\Rightarrow$ colour-sampling convergence tested when integrating $2\mathrm{Re}(\mathcal{M}^{(0)^*}\mathcal{M}^{(1)})$ over phase space ✓

Graphics processing units (GPUs) for

LO matrix-element evaluations

- *Introductory words*
- *GPU hardware & memory structure*
- *Monte Carlo program for LO leading-colour n-gluon MEs*
- *Results – timing, cross sections, distributions*

Some comments in the beginning

- Fast tree-level event generators are needed for multi-particle final states.
  - ⇒ evaluation time for event generation is crucial as one needs to average over many events to obtain good statistics for cross sections and observables
  - ⇒ not only @ LO, also @ NLO to calculate real-emission corrections ...
  - ⇒ ... and tree-level matrix elements when using generalized unitarity-cut methods to determine the virtual corrections
- Throw large computer farms/grids at the problem.
  - expensive; require certain infrastructure and maintenance
  - What if problem could be handled on a single, affordable PC ?
- Graphical Processor Units (GPUs) in addition to CPUs give an option. Explore capabilities.
  - ⇒ first applications within the framework of HELAS ME generator [HAGIWARA, KANZAKI, OKAMURA, RAINWATER, STELZER, arXiv:0909.5257, arXiv:0908.4403]
  - ⇒ can tame but not overcome factorial scaling of Feynman diagrammatic approach
- Define the project: LO LC n-gluon scattering cross sections. ⇒ Tools needed ...
  - unit-weight phase-space generator ... implementation of RAMBO [KLEISS, STIRLING, ELLIS]
  - strong-coupling evaluation, PDFs using LHAPDF and observables
  - $gg \to 2, \ldots, 10\ g$ MEs ... Berends–Giele ordered recursions, use threading to tame n^4-scaling

GPU hardware and programming principles

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- The C1060 Nvidia Tesla GPU is a plug-in card for your desktop. GPU has its own memory.
- The Tesla chip is designed for numerical applications and programmable in C/limited C++.
- The chip has 30 multi-processors (MPs), each comes with 1024 processors (threads).
  Each thread has an unique number (for I/O handling etc.).
  Threads essentially execute same processor instructions over different data
  (... can skip ahead and wait for other threads to catch up).
- Desirable: trivial parallelization (Monte Carlo algorithms: 1 event per thread).
  So, in principle we can run 30720 MC generators in parallel, each running N events
  ... a speed-up of 30000 !!
- Approach limited by amount of available fast-access memory.
  - off-chip slow-access memory: 4 Gb; use for I/O only ... transfer to and bin results on CPU
  - on-chip fast-access memory: only kbs; registers and shared memory

    ⇒ 16384 32-bit registers per MP; once assigned only seen by specific thread; temporary storage for function evaluations

    ⇒ 16384 bytes shared memory per MP; accessible to all threads

Memory layout

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- The n-gluon recursion relation needs n momenta and $n(n-1)/2$ currents for a total of $n(n+1)/2$ single precision 4-vectors.
  - Recursion relations are very suitable for GPU (memory efficient & algorithmically simple).
- We need $(4 \cdot 4)\ n(n+1)/2$ bytes of fast accessible memory per event.
- This means $16384/(8\ n(n+1))$ events per MP.
- One constraint though: implementation needs 35 registers per thread, i.e. 16384/35=468 threads are useable per MP.

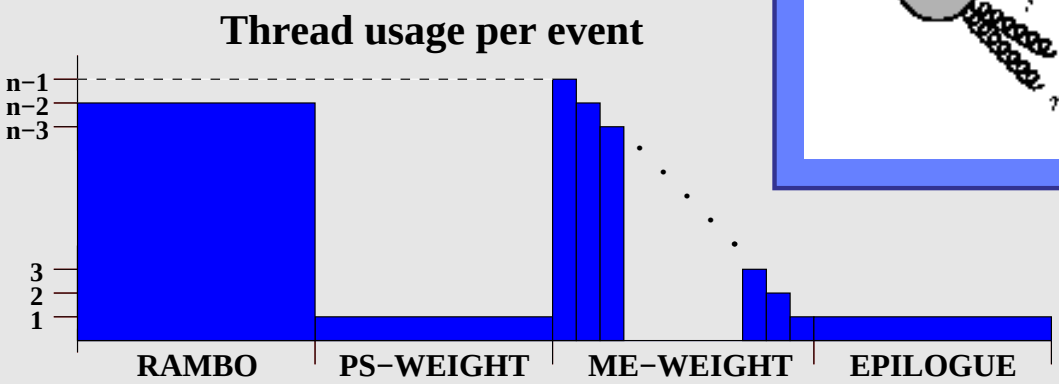
n	4	5	6	7	8	9	10	11	12
events per MP	102	68	48	36	28	22	18	15	13
avail. threads / evt	10	15	21	28	36	45	55	66	78
useable threads / evt	4	6	9	13	16	21	26	31	36

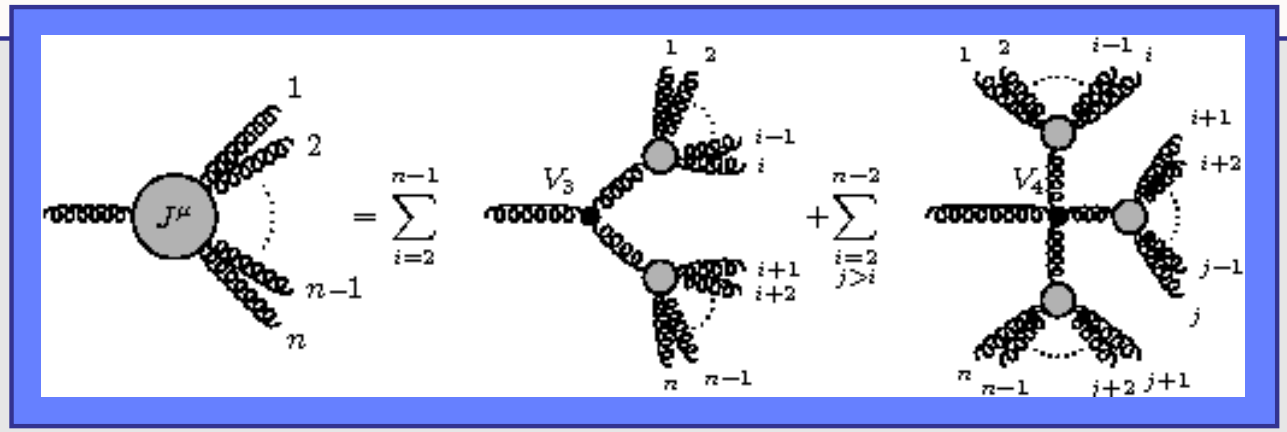
$\Rightarrow$ used threads per event $= n - 1$.

Processing events

[Giele, Stavenga, Winter, arXiv:1002.3446]

Tess MC program
(vircol.fnal.gov/TESS.html)





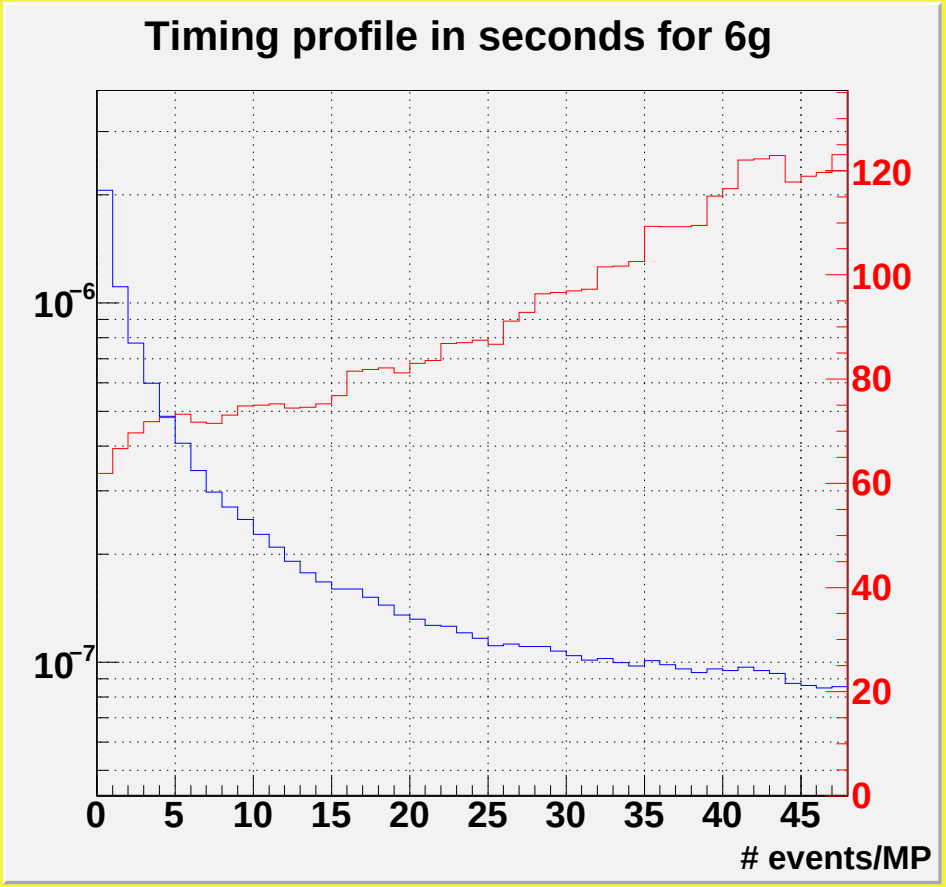
⇒ compute ME weight with BG recursion relation.

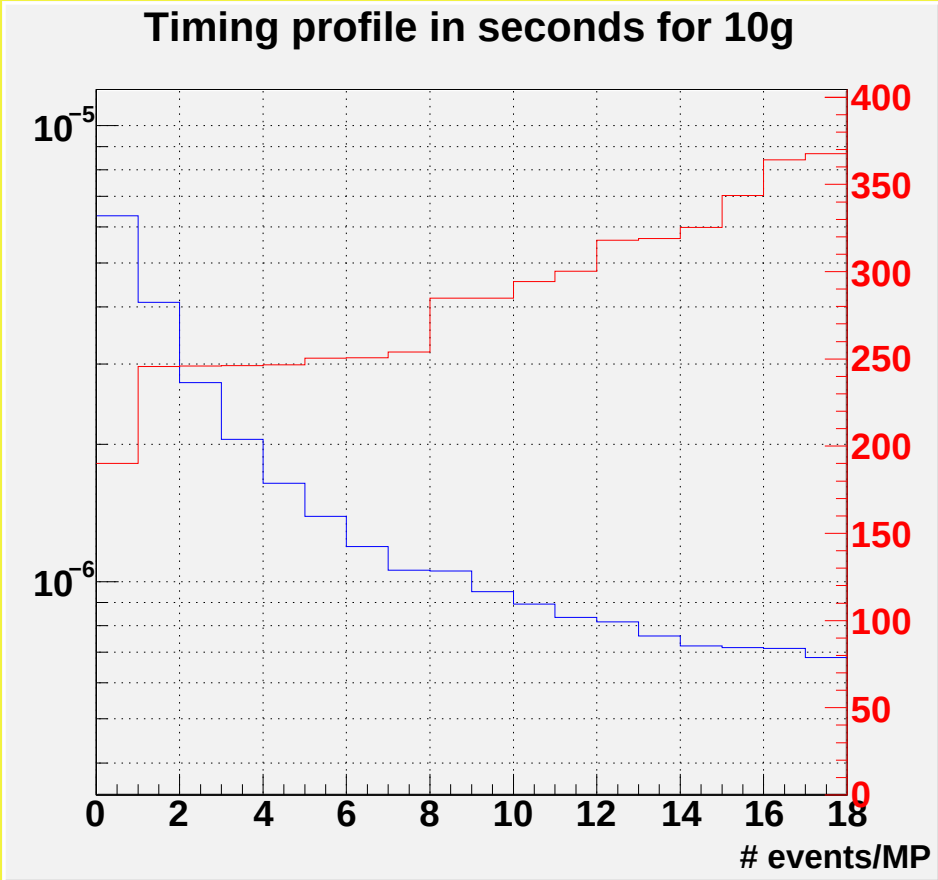
- Initialization phase: not shown.
- Rambo phase: $n-2$ threads to construct outgoing momenta.
- Phase-space weight: off-chip texture memory to store $\alpha_s(\mu)$ 1D and PDF $f_g(x,\mu)$ 2D grids.
  - ⇒ hardware performs linear interpolation between grid points for non-integer values
- ME weight: computed in $n-1$ steps instead of $n(n-1)/2$.
  - ⇒ reduces computational effort from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ complexity
  - ⇒ avoid complex multiplications; choose polarization vecs such that one has real-valued currents
  - ⇒ leading colour to avoid colour sum; use symmetry of FS (gluons only) to remove permutation sum over orderings

Computation times

[Giele, Stavenga, Winter, arXiv:1002.3446]

- Red curves: total GPU time in seconds to evaluate 10^6 sweeps (30×events/MP).
- Sweep time should be independent of #evts/MP, but queuing effects due to substantial amount of special-function calls.
- Blue curves: evaluation time per event = GPU time / total #evts.





⇒ best performance if max #evts available per MP.

Timing

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- Compare evaluation time per event on GPU with that of running the same algorithm on CPU [AMD Phenom(tm) II X4 940 (3 GHz)]. $P_n(m) = [(n-1)/n] \sqrt[m]{T_n/T_{n-1}}$
  - Speed-ups occur because events are evaluated in parallel.

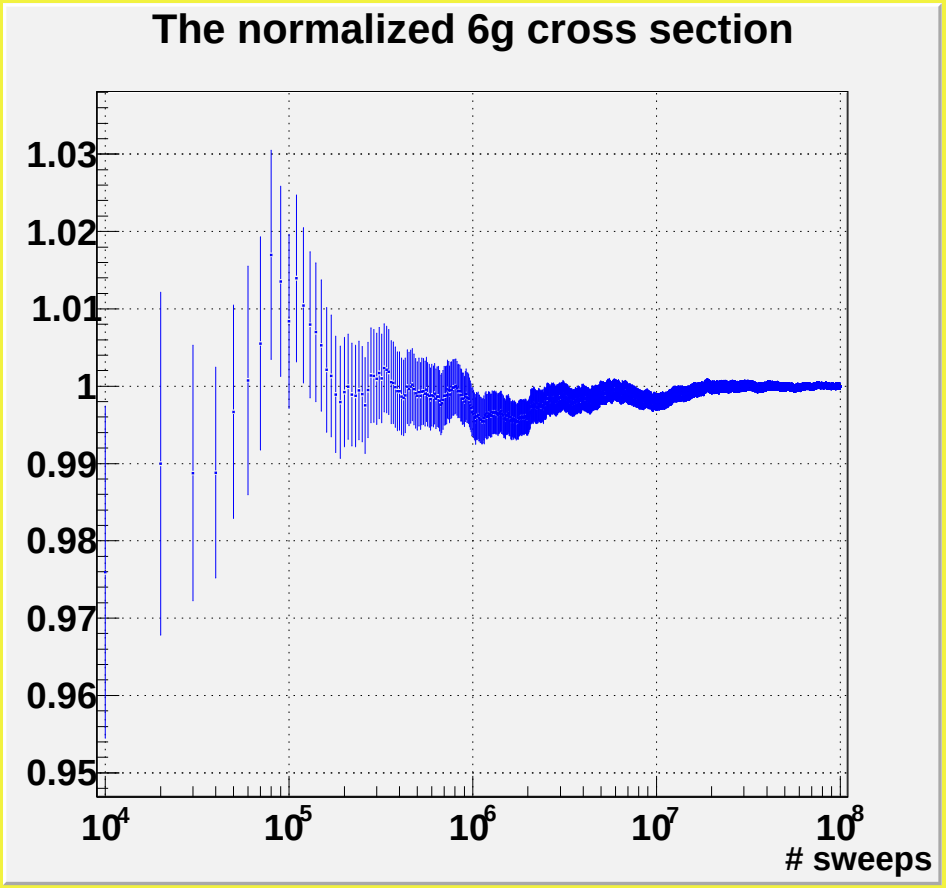
n	T_n^{GPU} (seconds)	$P_n(3)$	T_n^{CPU} (seconds)	$P_n(4)$	G_n
4	2.975×10^{-8}		8.753×10^{-6}		294
5	4.438×10^{-8}	0.91	1.247×10^{-5}	0.87	281
6	8.551×10^{-8}	1.03	1.966×10^{-5}	0.93	230
7	2.304×10^{-7}	1.19	3.047×10^{-5}	0.96	132
8	3.546×10^{-7}	1.01	4.736×10^{-5}	0.98	133
9	4.274×10^{-7}	0.94	7.263×10^{-5}	0.99	170
10	6.817×10^{-7}	1.05	1.044×10^{-4}	0.99	153
11	9.750×10^{-7}	1.02	1.529×10^{-4}	1.00	157
12	1.356×10^{-6}	1.02	2.129×10^{-4}	1.00	158

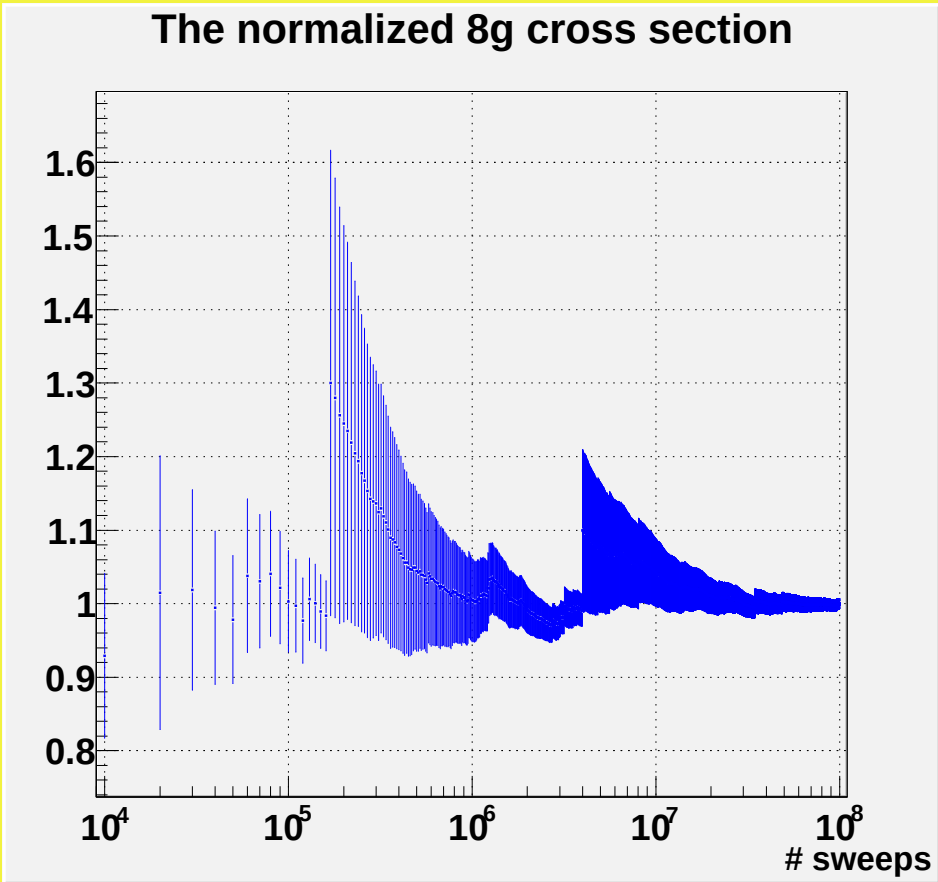
$\Rightarrow$ Gain.

LO LC multi-gluon cross sections

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- Convergence of cross section normalized to best xsec estimate.
- 14 TeV LHC, cteq6l1 PDFs, $\mu_{\rm F} = \mu_{\rm R} = M_Z$, $p_T^{\rm jet} > 20$ GeV, $|\eta^{\rm jet}| < 2.5$ & $\Delta R_{\rm jet-jet} > 0.4$.
- Subtleties: random-number generator cycle > total #evts; Kahan summation to avoid loss of precision (averaging $\mathcal{O}(10^{11})$ numbers).



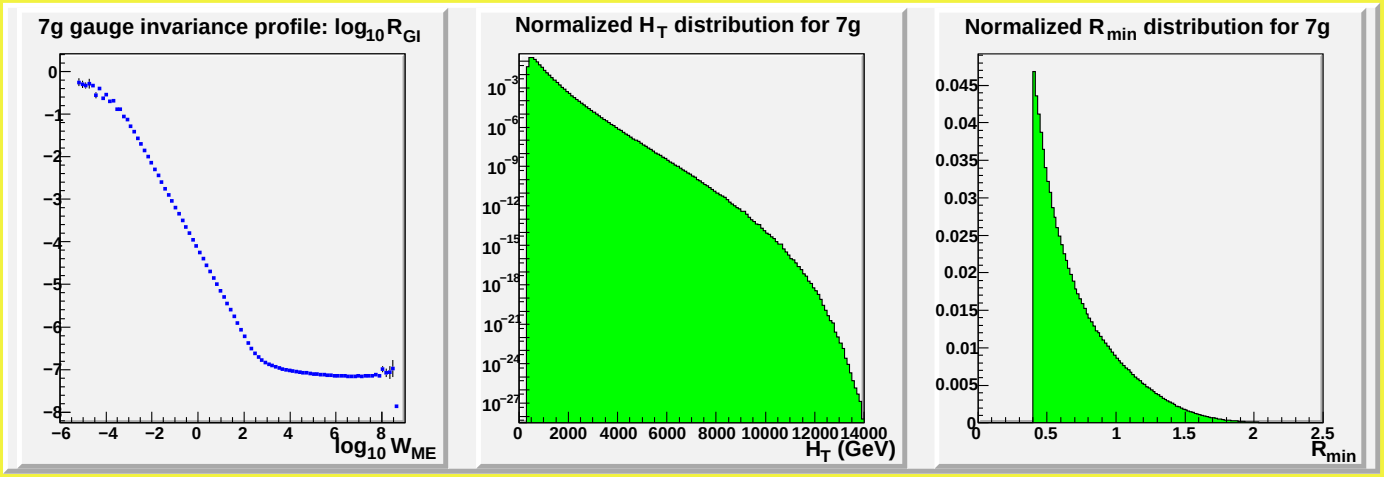


⇒ flat phase-space integration may under-sample regions of large weights → peaks.

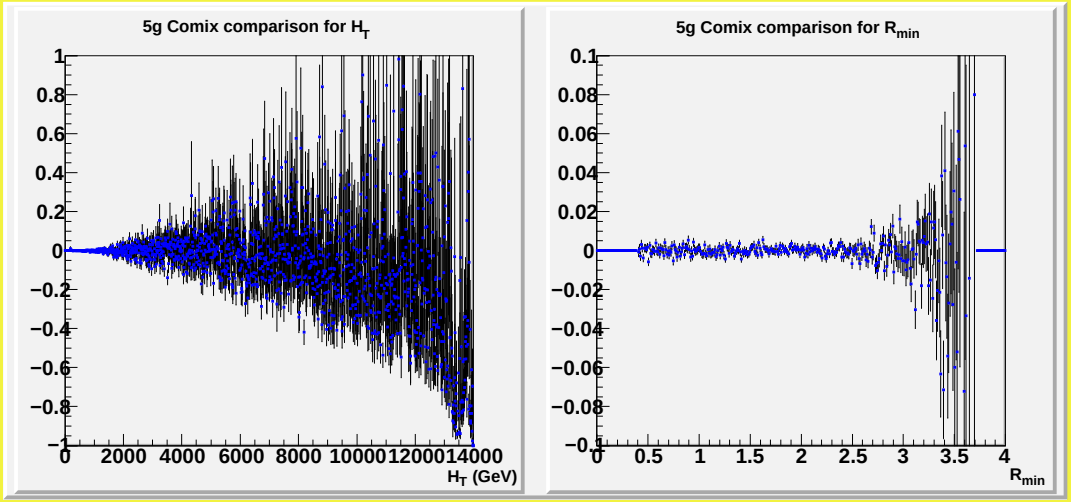
Distributions

[Giele, Stavenga, Winter, arXiv:1002.3446]

- Observables: H_T and minimum R-separation $R_{\min} = \min\{R_{ij}\}$ normalized to total xsec.
- 14 TeV LHC, cteq6l1 PDFs, $\mu_{\rm F} = \mu_{\rm R} = H_T$(upper), M_Z(lower), $p_T^{\rm jet} > 60$ GeV, $|\eta^{\rm jet}| < 2.0$ and $\Delta R_{\rm jet-jet} > 0.4$.
- upper: example for $gg \to 5g$; lower: $gg \to 3g$ compared to Comix. [Gleisberg, Höche]



Comix uses importance sampling, i.e. tail less populated, i.e. larger uncertainties @ large H_T.



Summary

- We obtained encouraging results in our first exploration of the potential of using multi-threaded GPU based workstations for Monte Carlo programming.
- Testbed chosen: leading-order leading colour n-gluon scatterings.

  $\Rightarrow$ **Tess** Monte Carlo program.
- Wrt the CPU based implementation, we found speed-ups ranging from $\mathcal{O}(300)$ and $\mathcal{O}(150)$ for 4-gluon and 12-gluon scattering, respectively.
- Outlook: @ LO – include quarks, vector bosons, subleading colour contributions, replace Rambo by Sarge. [Van Hameren, Papadopoulos]

  $\Rightarrow$ application to NLO
- GPU chips are still evolving rapidly ... next generation, Fermi chip (Fall 2010).

Thank you !!

for 3 wonderful years in FNAL's Theory Group.