HSF Generator WG: News and Plans

https://hepsoftwarefoundation.org/workinggroups/generators.html

A. Valassi, E. Yazgan, J. McFayden

HSG Generator WG Meeting #9 – 9th July 2020

https://indico.cern.ch/event/936481

News: paper

- WG paper uploaded on arxiv in April: https://arxiv.org/abs/2004.13687
- *A nice collaborative effort, many thanks to all authors and WG members!*
- Submitted to Springer 'Computing and Software for Big Science'
- Submitted to the LHCC review of HL-LHC computing in May 2020
- Submitted also to Snowmass 2021 (Computational Frontier CompF2)
  - Cross-posted by the organizers to the Theory Frontier TF07, too
  - Aside: Snowmass kickoff in August, join their mailing lists if interested!

News: LHCC review of HL-LHC computing (1)

- The first part of the review was held in May 2020
- A report is available as an appendix of the public minutes of the June 2020 LHCC meeting: https://cds.cern.ch/record/2719880
  - "The review panel appreciated the preparation of the documents and the presentations and found the interactions valuable."
  - "The experiments, WLCG, the DOMA project and HSF presented promising lists of R&D activities intended *to close the resource gap*, using the past experience that many changes can add up to a significant total."
  - "R&D activities include activities designed to *improve code performance on hardware architectures with accelerators (such as GPUs)* and to undertake infrastructure projects to integrate in High Performance Computing (HPC) centers. At this early stage, a multi-prong approach seems prudent. For the next review, there will be a more formal assessment of the gains expected […]"
  - "One area of concern shared by the experiments and WLCG is finding means to *ensure that the highly skilled personnel essential for R&D in computing and storage have meaningful career paths* within the LHC community to provide for sustainability and the need for continual evolution over the lifetime of HL-LHC."
  - "Common software has played an essential role for the community in the past and will do so, perhaps even more, in the future. *We note particularly that effort on generators is needed as one of the components to solve the HL-LHC computing challenge, however the required work does not fit into the established funding schemes.*"
  - "The HL-LHC Computing and Software Review committee anticipates a second review in 9-12 months which will focus on detailed R&D roadmaps."

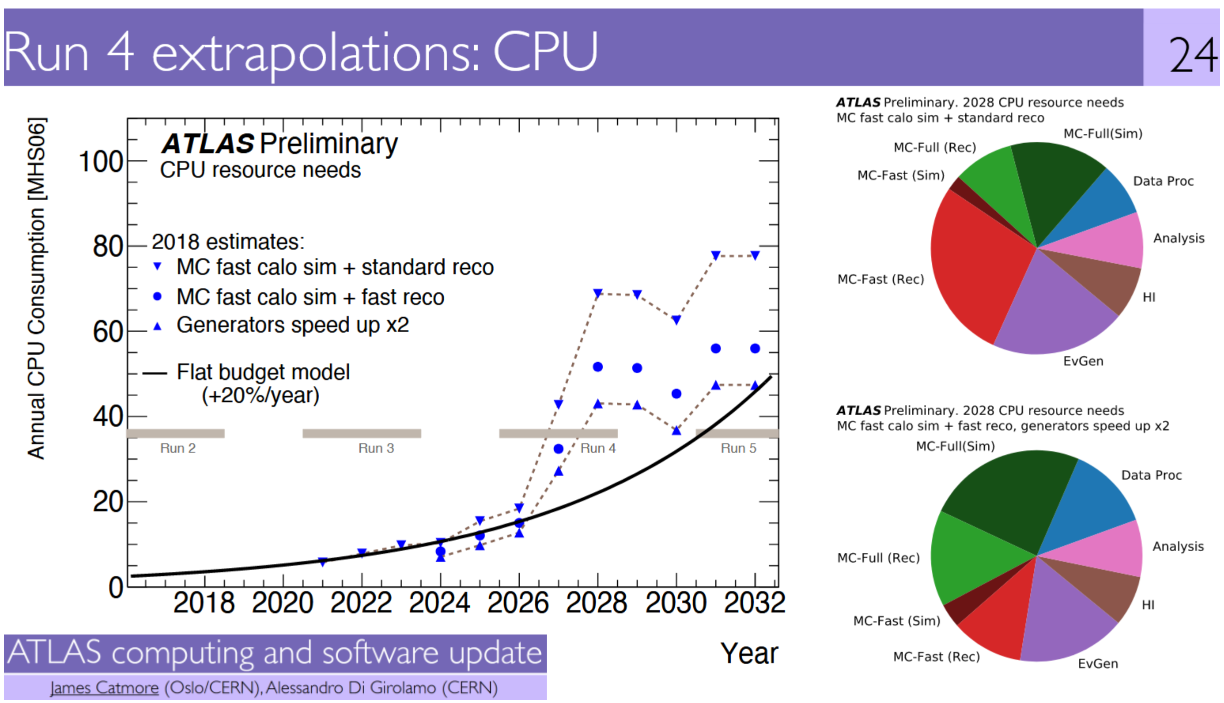
News: LHCC review of HL-LHC computing (2)

- The review will continue throughout 2021 and possibly beyond
- *Our WG has been asked to present "the status, progress, plans in the generators area" to the LHCC on 1st September 2020* (indico)
- Rest of this talk: recap of WG priorities as described in the LHCC paper
  - Several topics we would like to and plan to hear about in future WG meetings
  - A skeleton of what we could present in September to the LHCC
  - ***We need your feedback!*** About future meetings, and about the LHCC talk…

Generators and HL-LHC *computing*

- Main goal of our HSF WG is to deal with *software and computing issues*
  - Theoretical research is the <u>foundation</u> on which this work is based…
  - … but many in this WG are not theorists, or not even physicists!

- One of the main issues (not the only one!): *HL-LHC computing resource gap*



<u>WLCG meeting with LHCC referees, Feb. 2020</u>

In the case of ATLAS:

CPU cost of generators as a fraction of WLCG resources: in the ballpark of 10%-20%

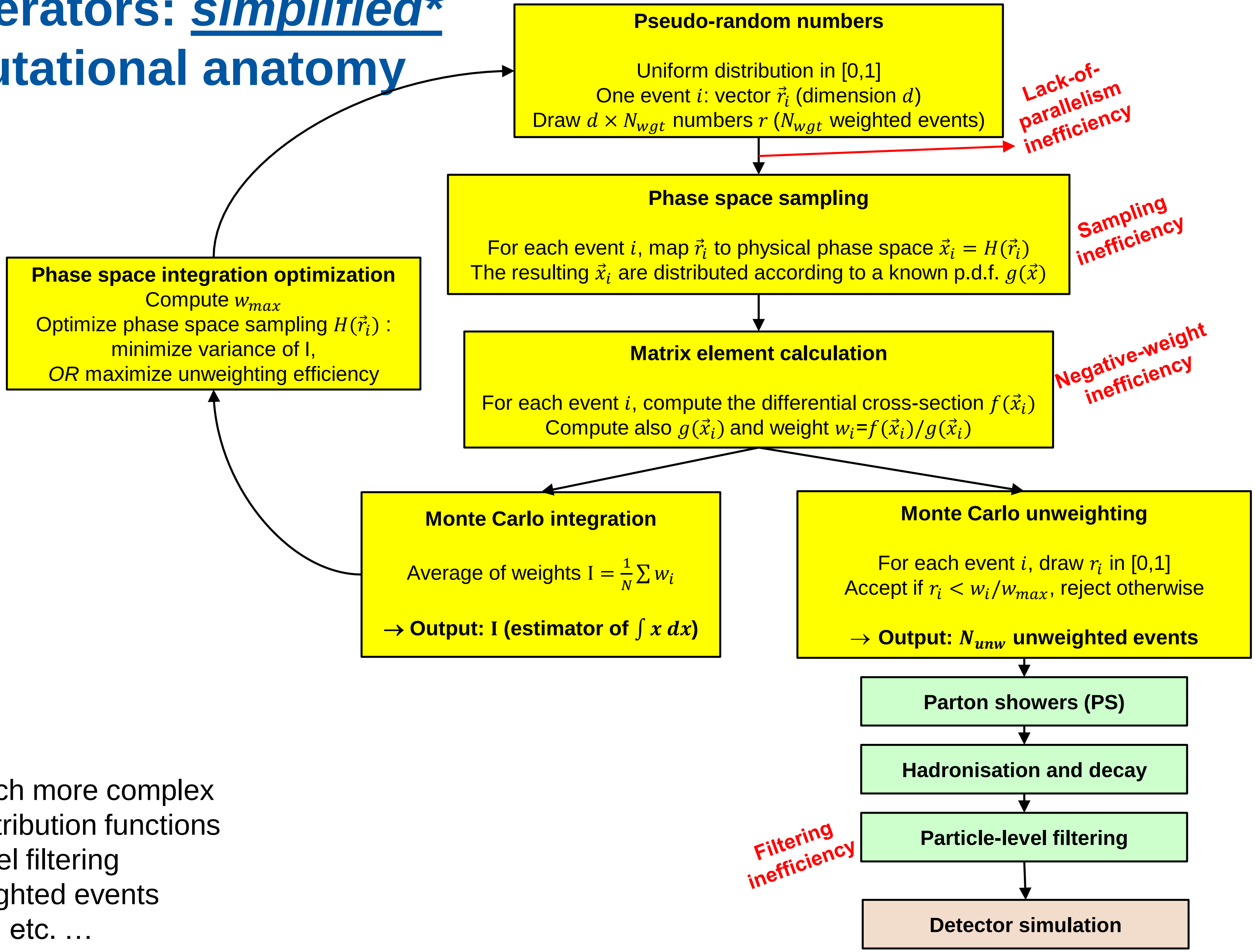
Overall generator speedup by a factor 2 is considered an R&D goal for HL-LHC by 2028

Main areas we want to explore

- Main priorities of the WG according to the conclusions of our paper:
  - 1. Gain a more detailed understanding of current CPU costs by accounting and profiling.
  - 2. Survey generator codes to understand the best way to move to GPUs and vectorized code, and prototype the port of the software to GPUs using data-parallel paradigms.
  - 3. Support efforts to optimize phase space sampling and integration algorithms, including the use of Machine Learning techniques such as neural networks.
  - 4. Promote research on how to reduce the cost associated with negative weight events, using new theoretical or experimental approaches.
  - 5. Promote collaboration, training, funding and career opportunities in the generator area.

- A few other very important areas
  - 6. Analyse filtering strategies and inefficiencies in the experiments.
  - 7. Understand and estimate future additional costs due to NNLO and increased precision
  - …

MC generators: ***simplified**** computational anatomy



*Reality is much more complex
- Parton distribution functions
- Parton-level filtering
- PS on weighted events
- … etc. etc. etc. …

Issue #2
Data-parallel paradigms (GPUs and vectorization)

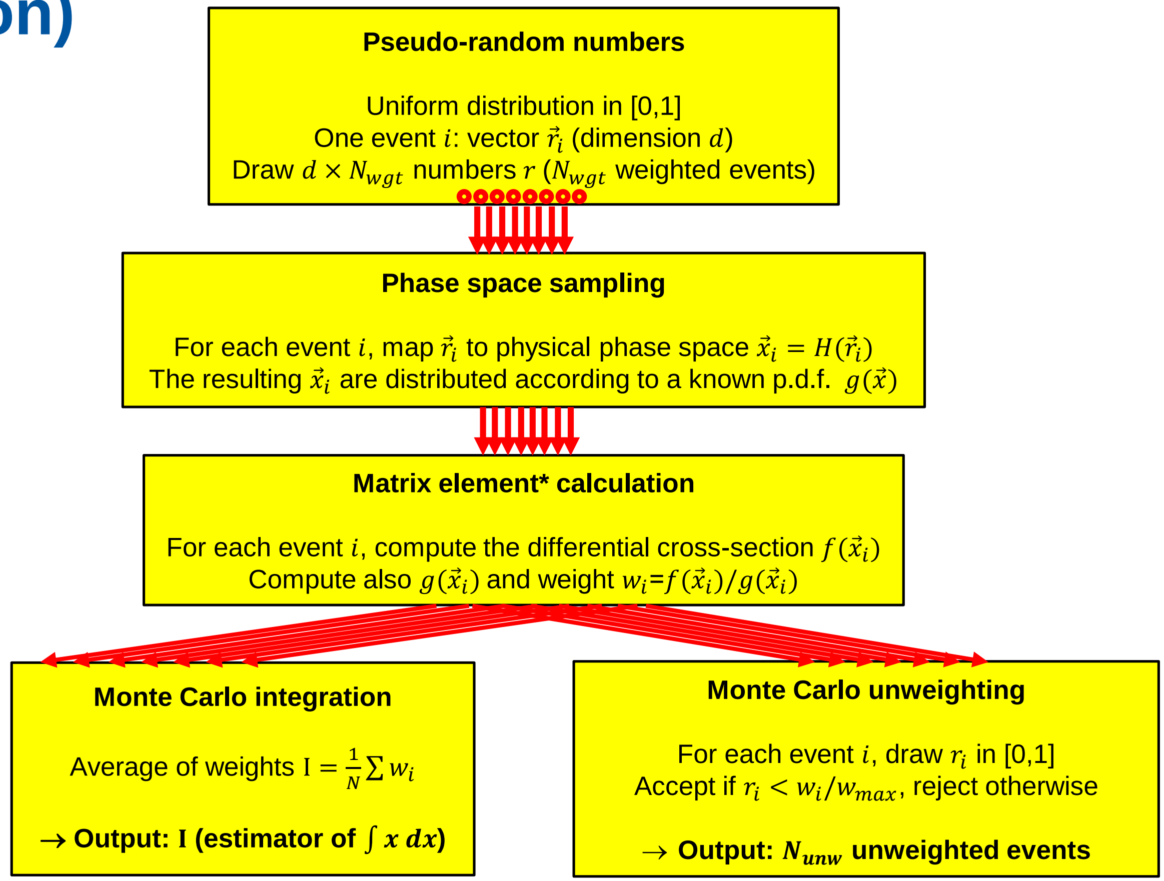
Generators should lend themselves naturally to **data-parallel paradigms**?

- **SPMD**: Single Program Multiple Data (GPU accelerators)
- **SIMD**: Single Instruction Multiple Data (CPU vectorization: AVX…)
- The computationally intensive part, the matrix element $f(\vec{x}_i)$, is ***the same function*** for all events i (in a given category of events)
- Unlike detector simulation (frequent if/then branches: on GPUs, branch divergence)

Potential interest of GPUs

- Faster (cheaper?) than on CPUs
- Exploit GPU-based HPCs

Work ongoing for MG5aMC on GPUs (Louvain, Argonne, CERN…): report on July 23?



**Note for software engineers: these calculations do involve some linear algebra, but "matrix element" does not refer to that! Here we compute one "matrix element" in the S-matrix (scattering matrix) for the transition from the initial state to the final state*

Issue #3: improving phase space *sampling algorithms*

Pseudo-random numbers

Uniform distribution in [0,1]
One event i: vector $\vec{r}_i$ (dimension d)
Draw $d \times N_{wgt}$ numbers r (N_{wgt} weighted events)

Phase space sampling

For each event i, map $\vec{r}_i$ to physical phase space $\vec{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{H}(\vec{\boldsymbol{r}}_{\boldsymbol{i}})$
The resulting $\vec{x}_i$ are distributed according to a known p.d.f. $g(\vec{x})$

Matrix element calculation

For each event i, compute the differential cross-section $f(\vec{x}_i)$
Compute also $g(\vec{x}_i)$ and weight w_i=$f(\vec{x}_i)/g(\vec{x}_i)$

Monte Carlo integration

Average of weights $\mathrm{I} = \frac{1}{N}\sum w_i$

→ **Output: I (estimator of $\int x\, dx$)**

Monte Carlo unweighting

For each event i, draw r_i in [0,1]
Accept if $r_i < w_i/w_{max}$, reject otherwise

→ **Output: N_{unw} unweighted events**

Phase space integration optimization
Compute w_{max}
Optimize phase space sampling $H(\vec{r}_i)$:
minimize variance of I,
***OR* maximize unweighting efficiency**

Work ongoing in many teams:
- J. Bendavid's talk Jan 2020
- Dedicated future meeting?

Unweighting efficiency is $\frac{N_{wgt}}{N_{unw}} = \frac{\langle w \rangle}{w_{max}}$
- The closer $g(\vec{x})$ is to $f(\vec{x})$, the better
- NB: maximizing efficiency related to, but not the same as, minimizing Var(I)

Many techniques for sampling
- Importance, stratified, adaptive…
- Multi-channel
- Machine Learning, normalizing flows…

Example: *Sherpa* W+jets
- **Efficiency is ~30% for W+0jets**
- **Efficiency is ~0.08% for W+4jets**

Unweighting efficiency $\langle w \rangle / w_{\max}$		LO QCD $n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$W^+ + n$ jets	SHERPA	2.8×10^{-1}	3.8×10^{-2}	7.5×10^{-3}	1.5×10^{-3}	8.3×10^{-4}
	NN + NF	6.1×10^{-1}	1.2×10^{-1}	1.0×10^{-2}	1.8×10^{-3}	8.9×10^{-4}
	Gain	2.2	3.3	1.4	1.2	1.1

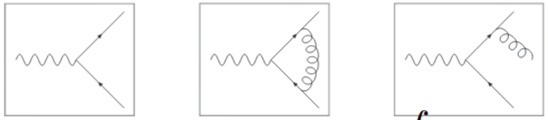
https://doi.org/10.1103/PhysRevD.101.076002

Issue #4: reduce the cost of *negative-weight events* (only NLO and beyond)

MC@NLO: https://doi.org/10.1088/1126-6708/2002/06/029
Matching NLO QCD and parton showers (avoid double counting)

Marco Zaro – https://cp3.irmp.ucl.ac.be/projects/madgraph/wiki/Pavia2015

B, V, R: matrix elements
MC: parton shower



$$d\sigma^n_{NLO} = d\sigma^n_{LO} + d\sigma^n_V + \int d\Phi_1\, d\sigma^{n+1}_R$$

$$\frac{d\sigma_{\text{“MC@NLO”}}}{dO} = \underbrace{\left[\int d\Phi_n\left(B+V+\int d\Phi_1 MC\right)\right] I^n_{MC}(O)}_{\text{S-events}} + \underbrace{\left[\int d\Phi_{n+1}(R-MC)\right] I^{n+1}_{MC}(O)}_{\text{H-events}}$$

S and **H** **events**: two separate sets of events (different matrix elements)
Integral = S+H is positive – but individual events can have negative weights

	MC@NLO			MC@NLO-Δ		
	111	221	441	Δ-111	Δ-221	Δ-441
$pp \to e^+e^-$	6.9% (1.3)	3.5% (1.2)	3.2% (1.1)	5.7% (1.3)	2.4% (1.1)	2.0% (1.1)
$pp \to e^+\nu_e$	7.2% (1.4)	3.8% (1.2)	3.4% (1.2)	5.9% (1.3)	2.5% (1.1)	2.3% (1.1)
$pp \to H$	10.4% (1.6)	4.9% (1.2)	3.4% (1.2)	7.5% (1.4)	2.0% (1.1)	0.5% (1.0)
$pp \to Hb\bar{b}$	40.3% (27)	38.4% (19)	38.0% (17)	36.6% (14)	32.6% (8.2)	31.3% (7.2)
$pp \to W^+j$	21.7% (3.1)	16.5% (2.2)	15.7% (2.1)	14.2% (2.0)	7.9% (1.4)	7.4% (1.4)
$pp \to W^+t\bar{t}$	16.2% (2.2)	15.2% (2.1)	15.1% (2.1)	13.2% (1.8)	11.9% (1.7)	11.5% (1.7)
$pp \to t\bar{t}$	23.0% (3.4)	20.2% (2.8)	19.6% (2.7)	13.6% (1.9)	9.3% (1.5)	7.7% (1.4)

Table 1: Fractions of negative-weight events, f, and the corresponding relative costs, $c(f)$ (in round brackets), for the processes in eqs. (5.7)–(5.13), computed with MC@NLO (columns 2–4) and with MC@NLO-Δ (columns 5–7), for three different choices of the folding parameters.

https://arxiv.org/abs/2002.12716

Pseudo-random numbers

Uniform distribution in [0,1]
One event i: vector $\vec{r}_i$ (dimension d)
Draw $d \times N_{wgt}$ numbers r (N_{wgt} weighted events)

↓

Phase space sampling

For each event i, map $\vec{r}_i$ to physical phase space $\vec{x}_i = H(\vec{r}_i)$
The resulting $\vec{x}_i$ are distributed according to a known p.d.f. $g(\vec{x})$

↓

Matrix element calculation

For each event i, compute the differential cross-section $f(\vec{x}_i)$
Compute also $g(\vec{x}_i)$ and weight w_i=$f(\vec{x}_i)/g(\vec{x}_i)$

↓

Monte Carlo unweighting

For each event i, draw r_i in [0,1]
Accept if $r_i < w_i/w_{max}$, reject otherwise

→ **Output: N_{unw} unweighted events**

↓

Parton showers (PS)

Work ongoing in many teams:
- Dedicated future meeting?

For a fraction r of negative weight events:
Need a factor $\frac{1}{(1-2r)^2}$ more events to generate, simulate, reconstruct
Example for $Hb\bar{b}$: r~40% implies ~25 times as many events!

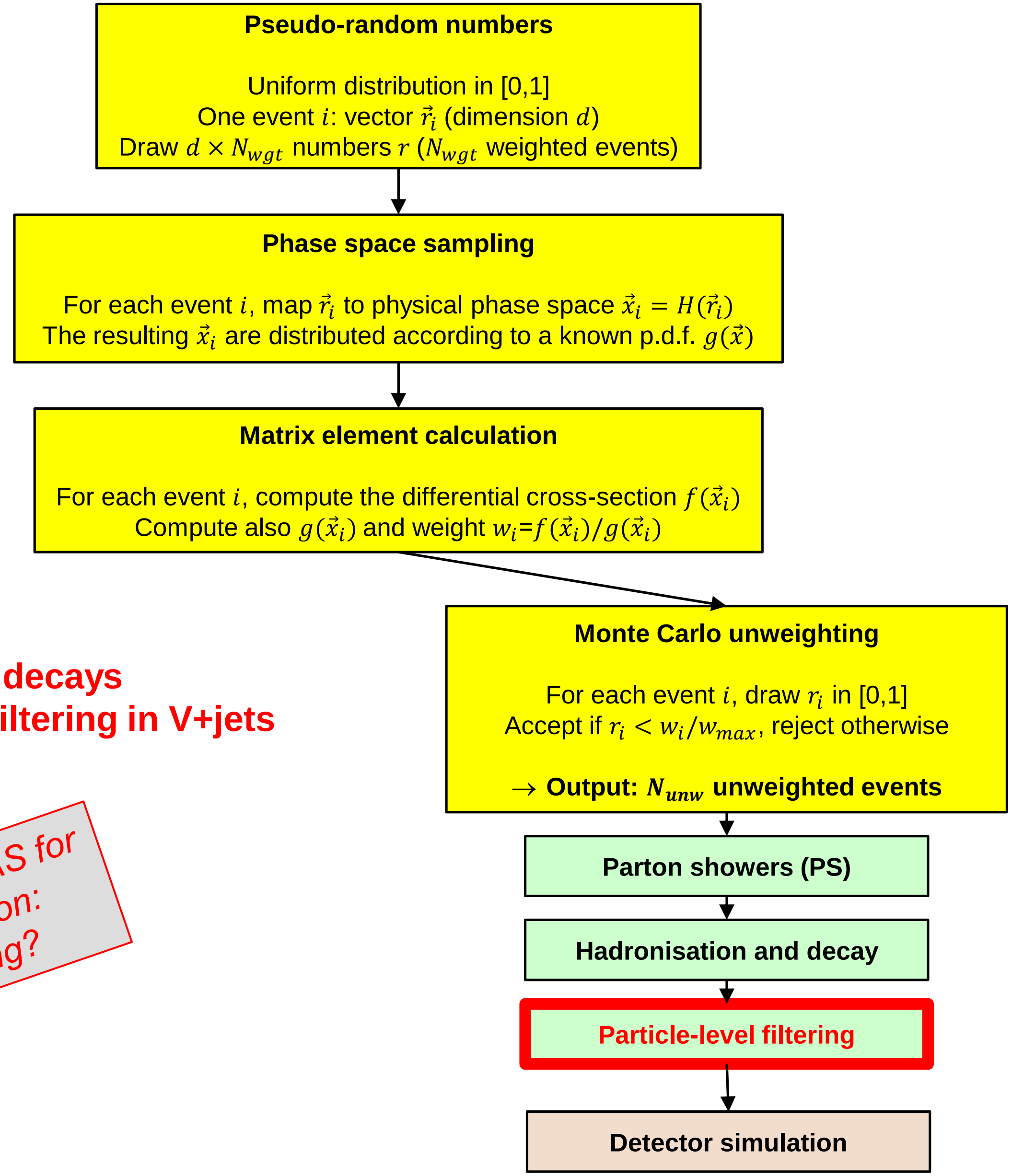
Issue #6:
filtering inefficiencies

For some processes, in the experiments:
- Generate large inclusive samples
- Filter on final state criteria

Possible improvements have been suggested:
- Develop filtering tools within the generators
- Filtering one production into many streams

Examples
- **CMS: ~0.01% efficiency for specific Λ_B decays**
- **ATLAS: ~10% efficiency for B-hadron filtering in V+jets**

Interest in LHCb, CMS, ATLAS for a cross-experiment discussion:
- Dedicated future meeting?



Issue #1: *accounting and profiling*

1. Accounting of ATLAS and CMS

- A lot of work in 2019 (see table 1)
- ATLAS update Jan 2020: HS06 seconds
- *CMS update Jan 2020: separate figures will be available for GEN and SIM in the future, wait for new productions and then report*

It may be interesting to understand, per sample, also the sampling inefficiency, filtering inefficiency, fraction of negative weights, merging inefficiency (for multi-leg setups)…

Exp		ATLAS			CMS		
Process	Prec	Gen	nEvts	CPU[s]	Gen	nEvts	CPU[s]
V incl	NLO	PW+Py8	1175M	0.6B			
V+jets	LO	MG5aMC+Py8	445M	33B	MG5aMC	1618 M	3.1B
V+jets	NLO	Sherpa	1070M	225B	MG5aMC	4578 M	44.4B
$t\bar{t}$+jets	LO				MG5aMC	430 M	26.4B
$t\bar{t}$ incl	NLO	PW+Py8/MG5aMC+Py8	2040M	8B	PW[1]	1940 M	4.7B
Diboson	NLO	Sherpa/PW+Py8	416M	50B	PY/PW/MG/HW	712 M	5.6B
γ+jets	LO	Sherpa	64M	2B			
γ+jets	NLO	Sherpa	33M	11B			
ttV	NLO	MGaMC+Py8	40M	0.1B	MG5aMC	154 M	2.0B
single top	NLO	PW+Py8/MG5aMC+Py8	328M	4B	PW[2]	880 M	2.1B
multijet	[N]LO	Sherpa/Pythia8	825M	95B	Pythia8/MG5aMC	1950 M	7.2B
TOTAL		event generation	6.4B	428B		12.3B	96.9B
		simulation	7.2B	1515B		–	516.1B
		reconstruction	8.0B	550B		–	429.1B

Table 1: Summary of number of events produced in the MC campaign corresponding to the 2017 data-taking split by the dominant backgrounds and given for ATLAS and CMS. The total number of events and CPU consumption for all samples in this campaign are also given. Note CMS numbers here use HS06.s instead of second. Moreover, CMS Simulation is relatively faster due to many improvements as mentioned here.

Accounting: last update at the HSF generator WG June 2019 meeting

Suggestions for WG work and future meetings:
- Update on CMS accounting
- Resume Sherpa vs MG5aMC profiling
- Discussion on scale parameters?

2. Profiling, e.g. Sherpa vs MG5aMC

- Early comparisons in 2019 (see figure 3)
- No reproducible setups yet – would be useful
- Email discussions on scale parameters in 2020

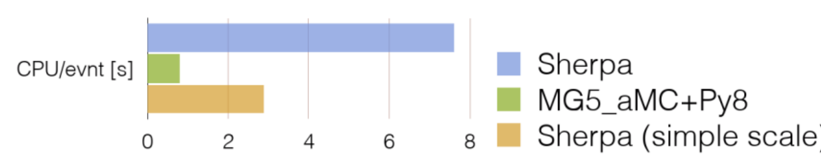


Figure 3: CPU per event $W \to e\nu$+0-2jets@NLO (using pre-integration grids).

Profiling: report at the HSF generator WG March 2019 meeting

Issue #5: *collaboration, funding and careers*

For instance:
no obvious career recognition for theorists in speeding up the code

Some may say
this is technical work,
not academic research

No simple solution

Raising awareness of this issue
with review bodies and funding agencies
is the first thing we can do

CERN/LHCC-2020-008
LHCC-142
June 2020

LARGE HADRON COLLIDER COMMITTEE

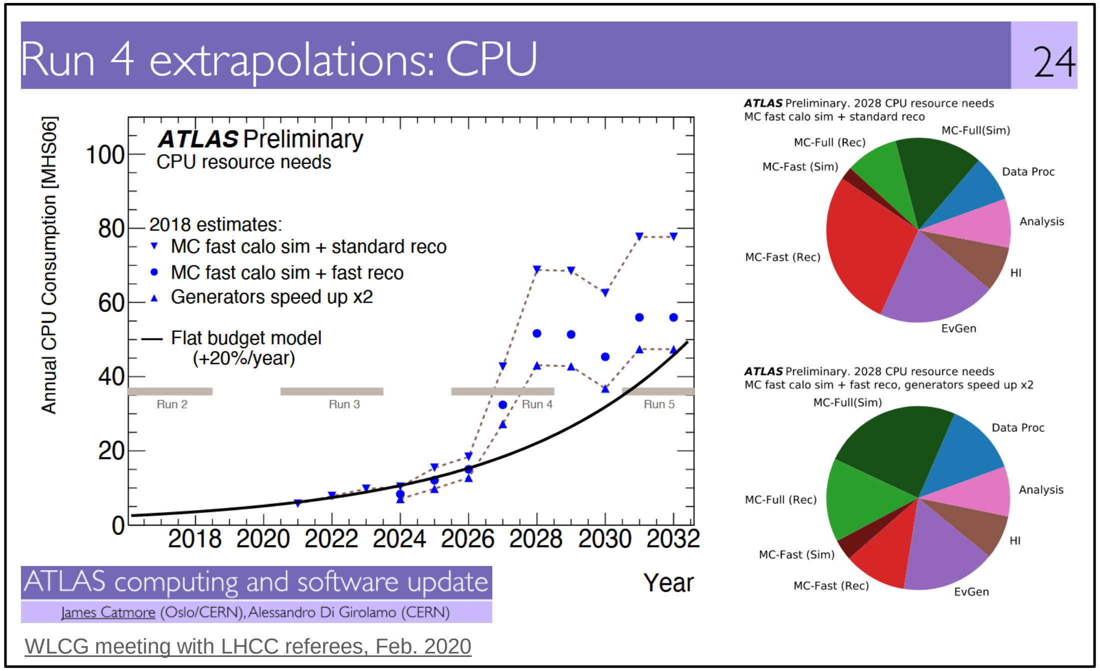
Minutes of the one-hundred-and-forty-second meeting held on

Thursday and Friday, 4-5 June 2020

HSF: The committee congratulates the HSF for establishing a forum where common software developments and techniques are discussed, especially for common software that extends beyond the LHC experiments. The value of this is recognized by the experiments and the community. Common software has played an essential role for the community in the past and will do so, perhaps even more, in the future. We note particularly that effort on generators is needed as one of the components to solve the HL-LHC computing challenge, however the required work does not fit into the established funding schemes.

Do you have specific suggestions?

Issue #7:
the future cost of improved precision



MC generators, towards HL-LHC:
- Larger data volumes
- Higher precision: higher jet multiplicities
- Higher precision: more NLO, more NNLO

How much more would NNLO calculations cost?
- More Feynman diagrams (slower calculations)
- Two-loop diagrams (more complex, more expensive)
- Higher fraction of negative weights?

How much NNLO would be required, and where?
- (and which NNLO calculations will be available?!)

Work ongoing in many teams:
- Talk by Massimiliano Grazzini today
- Dedicated future meeting?

Conclusions: next steps

- We are planning to have a meeting in two weeks on July 23rd
  - One suggested topic so far: report on the MG5aMC work on GPUs
  - Please let us know if you would like to present something!

- **Please give us your feedback! Thanks in advance! ☺**
  - On the content and format of next meetings:
    - Topical meetings dedicated to one specific issue?
    - Which topics would you like the WG to discuss first?
  - On any specific work you would like to do within the WG
  - On specific messages to pass (or not) to the LHCC in September

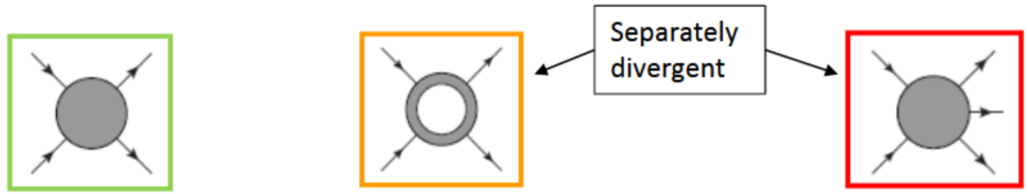
Backup slides

Gionata Luisoni – https://indico.cern.ch/event/602457/contributions/2435408

Structure of a NLO calculation

27/03/2017 - Gionata Luisoni

- NLO calculation for an observable O consists of different ingredients:



$$\langle O \rangle = \int O\, d\sigma = \int d\mathbf{\Phi}_n\, O(\mathbf{\Phi}_n)\left[B(\mathbf{\Phi}_n) + V_b(\mathbf{\Phi}_n)\right] + \int d\mathbf{\Phi}_n\, d\Phi_r\, O(\mathbf{\Phi}_n, \Phi_r)\, R(\mathbf{\Phi}_n, \Phi_r)$$

- [Parametrized (n+1)-body phase space $\mathbf{\Phi}_{n+1}$ in terms of Born and radiation $\mathbf{\Phi}_{n+1} = \{\mathbf{\Phi}_n, \Phi_r\}$]

- Divergent parts can be regulated e.g. with a subtraction scheme:

$$\begin{aligned} \langle O \rangle &= \int d\mathbf{\Phi}_n\, O(\mathbf{\Phi}_n)\left[B(\mathbf{\Phi}_n) + V_b(\mathbf{\Phi}_n) + \int d\Phi_r\, C(\mathbf{\Phi}_n, \Phi_r)\right] \\ &+ \int d\mathbf{\Phi}_n\, d\Phi_r\, \underbrace{\left[O(\mathbf{\Phi}_n, \Phi_r)\, R(\mathbf{\Phi}_n, \Phi_r) - O(\mathbf{\Phi}_n)\, C(\mathbf{\Phi}_n, \Phi_r)\right]}_{\text{finite}} \end{aligned}$$

- Defining: $V(\mathbf{\Phi}_n) = V_b(\mathbf{\Phi}_n) + \int d\Phi_r\, C(\mathbf{\Phi}_n, \Phi_r) \quad \Longleftarrow \text{finite}$

$$\langle O \rangle = \int d\mathbf{\Phi}_n\, O(\mathbf{\Phi}_n)\left[B(\mathbf{\Phi}_n) + V(\mathbf{\Phi}_n)\right] + \int d\mathbf{\Phi}_n\, d\Phi_r\left[O(\mathbf{\Phi}_n, \Phi_r)\, R(\mathbf{\Phi}_n, \Phi_r) - O(\mathbf{\Phi}_n)\, C(\mathbf{\Phi}_n, \Phi_r)\right]$$

Aside – about "matrix elements"

- What is a "matrix" to a software engineer and to a theoretical physicist?
  - Language is in itself a challenge in a multi-domain collaboration!
- Software engineers speak of (and like!) matrix algebra calculations
  - Matrix algebra maps naturally to data parallel paradigms (SIMD/vectorization, SPMD/GPUs): the same operation (add/multiply) is repeated in a loop
  - The LINPACK benchmark (e.g. for ranking Top500 supercomputers) computes the LU factorization of a dense matrix as the product of two triangular matrices
- Theoretical physicists speak of the scattering matrix (S-matrix)
  - The probability amplitude (invariant amplitude) for the quantum transition from an initial state |i> to a final state |f> is the "matrix element" $S_{fi} = \langle f|S|i \rangle$
  - Using Feynman diagram, physics event generators compute the contribution to the matrix element S_{fi} at a given order in its perturbative expansion
- Data parallelism (GPUs, vectorization) actually is a good fit for MC event generators, but the reason is NOT that they compute "matrix elements"!
  - Rather: same function computed over many phase space points, no branching