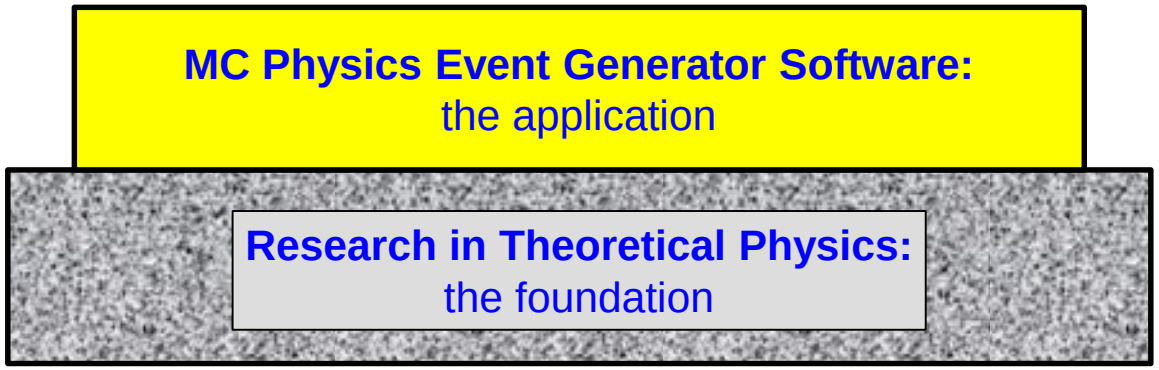
Monte Carlo generators challenges and strategy towards HL-LHC

A. Valassi, E. Yazgan, J. McFayden

On behalf of the HSF generator WG – https://hepsoftwarefoundation.org/workinggroups/generators.html

LHCC meeting, 1st September 2020 - https://indico.cern.ch/event/877842/

MC generators and HL-LHC: foreword



A multi-disciplinary challenge

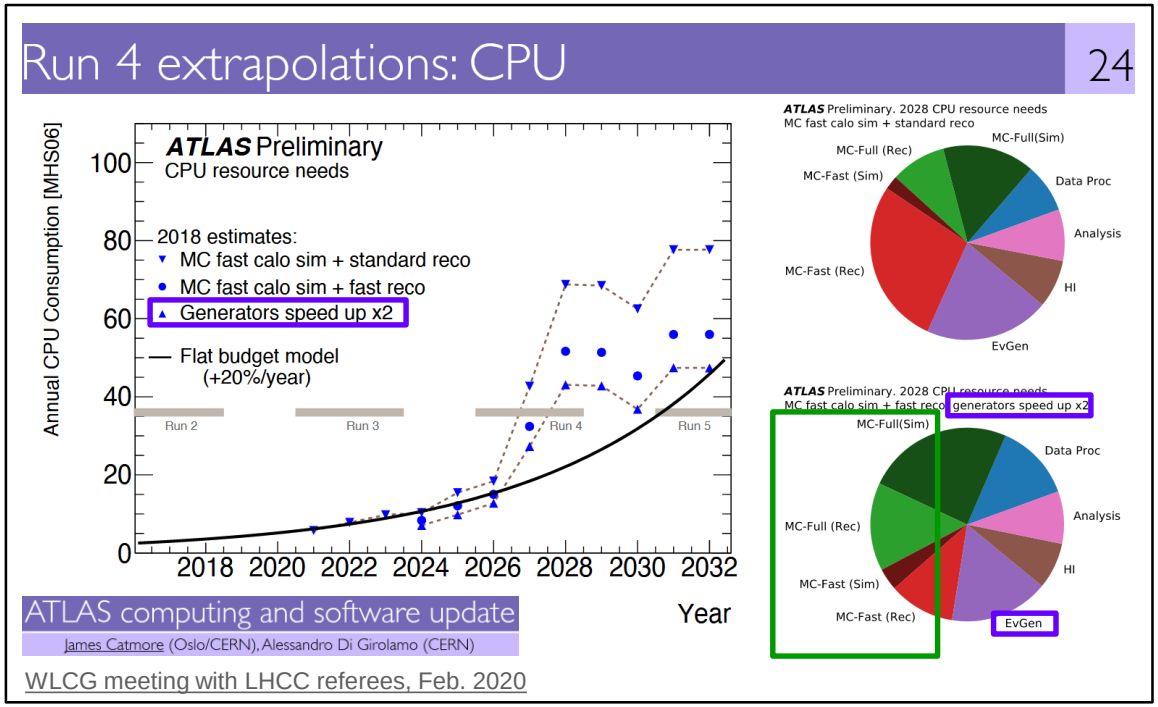
A diverse mix of people/skills

Theorists
Experimentalists (physics)
Experimentalists (computing)
Software engineers

- *This talk focuses on the **Software** and **Computing** aspects of MC generators*
  - I only discuss the physics issues and choices which have a direct impact on S&C
- This is the same general goal of the HSF physics event generator WG
  - The WG was formed after the HSF generator S&C workshop at the end of 2018
  - A diverse mix of people/skills: theorists, experimentalists, software engineers…
- This talk is based on our WG paper: https://arxiv.org/abs/2004.13687
  - Contributed to the LHCC review, submitted to CSBS and to Snowmass 2021
  - *I will focus on issues relevant to ATLAS and CMS (apologies to LHCb and ALICE)*

MC generators and *HL-LHC software and computing*

- One of the main issues (not the only one!): *HL-LHC computing resource gap*
  - Generator performance must also keep up with higher physics precision



CPU cost of generators as a fraction of WLCG CPU resources: for ATLAS, ballpark of 10%-20% (for CMS, this is lower)

ATLAS considers an overall generator speedup by a factor x2 as an R&D goal for HL-LHC

Side note: the higher the fraction of **negative-weight events from MC generators**, the higher the CPU cost of MC simulation, MC reconstruction and analysis (need more MC events)

- Many other challenges, including:
  - WLCG software workloads on non-traditional resources (HPCs, GPUs…)
  - Funding and careers (especially at the theory/experiment/computing interface)

MC generators in the LHCC review of HL-LHC computing

- Some of the challenges above are common to other areas of HEP S&C
- This is well captured in the first report of the LHCC review of HL-LHC computing: https://cds.cern.ch/record/2725487
  - "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 *develop and 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."
  - "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. […] Training and education are an important part of the strategy."
  - "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.*"
  - "Starting with the performance of the current software stack and existing computing model, *ATLAS and CMS should produce tables showing how they expect CPU and storage requirements to be mitigated by current and future developments, within and outside of ATLAS and CMS*, and assess the status and risks associated with each of these developments. These tables would then underpin a credible strategy."
- This talk provides more details about *specific* challenges and strategies for MC physics event generators towards HL-LHC

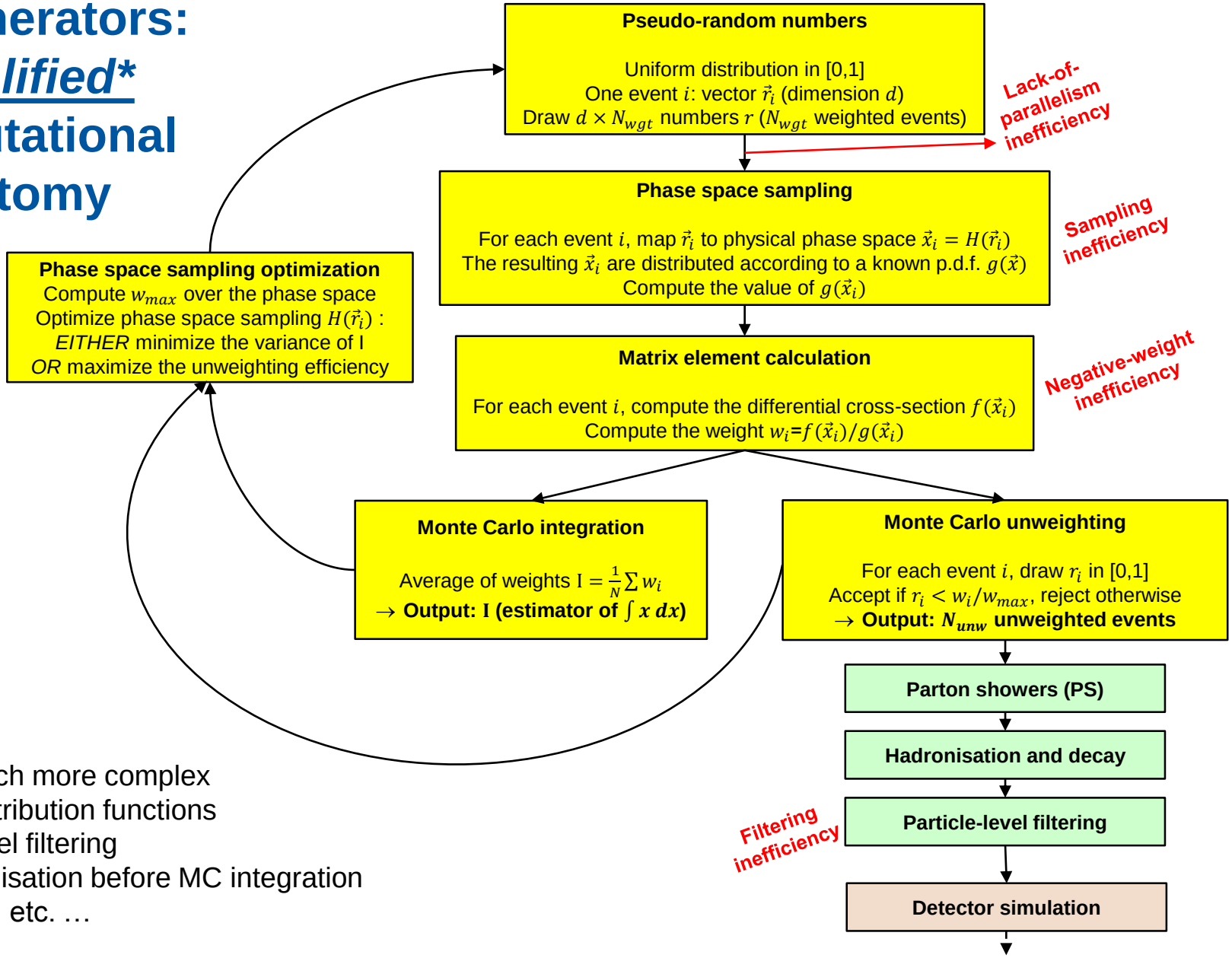
Executive summary: challenges and plans for MC event generators

- Main WG priorities as discussed in our paper: (same level, not an ordered list)
  - 1. Gain a better understanding of current CPU costs by accounting and benchmarking
  - 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 (also discussed in detail in the paper):
  - 6. Analyse filtering strategies and inefficiencies in the experiments
  - 7. Understand and estimate future additional costs due to NNLO and increased precision
  - 8. Other CPU speedup opportunities via software technicalities or algorithmic improvements

- *There is not a single major issue: we need a multi-prong strategy*
  - *Many different challenges → Many different opportunities for improvement and cost savings*

MC generators: *simplified** computational anatomy



*Reality is much more complex
- Parton distribution functions
- Parton-level filtering
- PS/hadronisation before MC integration
- … etc. etc. etc. …

A complex and heterogeneous problem

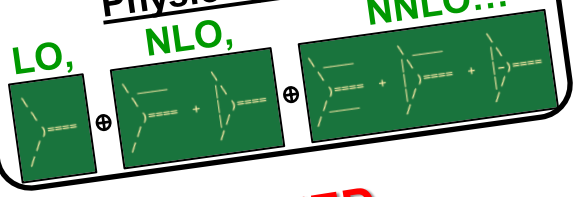
Sampling algorithms:
Vegas, Miser, Rambo, Bases/Spring, Mint, Foam, Vamp, MadEvent, Comix…

Generators:
MadGraph5_aMC@NLO (MG5aMC), Sherpa, Powheg, Pythia, Herwig, Alpgen…

LHC final states:
V (W or Z boson) + jets, di-boson, ttbar, single top, ttV, multi-jet, gamma + jets…

Parton distribution functions:
LHAPDF,…

Physics precision:
LO, NLO, NNLO…



MC Physics Event Generator Software:
the application

Research in Theoretical Physics:
the foundation

AN EXTREMELY VARIED SOFTWARE (and use case) LANDSCAPE!

Matching and Merging prescriptions:
aMC@NLO, Powheg, KrkNLO, CKKW, CKKW-L, MLM, MEPS@NLO, MINLO, FxFx, UNLOPS, Herwig7 Matchbox..

Hadronization and Parton Showers:
Pythia, Herwig, Ariadne…

- Software (and theory) diversity is good for physics
  - It provides cross-checks and healthy competition
- But it complicates the definition of an R&D strategy
  - Many software packages to optimize (and maintain!)
  - Prioritization (“profiling”): is there a CPU “hotspot”?

Issue #1:
accounting / benchmarking

1. Accounting of ATLAS and CMS

NB: preliminary data (WIP) provided by ATLAS/CMS MC conveners

- 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*
- *CMS accounting is complicated by the fact that monitoring data is only kept for 18 months*

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

It would be useful to know, per sample, also

- the sampling inefficiency
- the filtering inefficiency
- the fraction of negative weights
- the merging inefficiency (for multi-leg setups)
- …

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

2. Benchmarking, e.g. Sherpa vs MG5aMC

- Early comparisons in 2019 (see figure 3)
- Focus on one main consumer: V+jets@NLO
- No reproducible setups yet – would be useful
- Email discussions on scale parameters in 2020

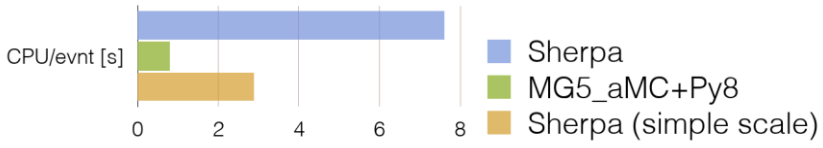


Figure 3: CPU per event $W \to e\nu$+0-2jets@NLO (using pre-integration grids).

Benchmarking: report at the HSF generator WG March 2019 meeting

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

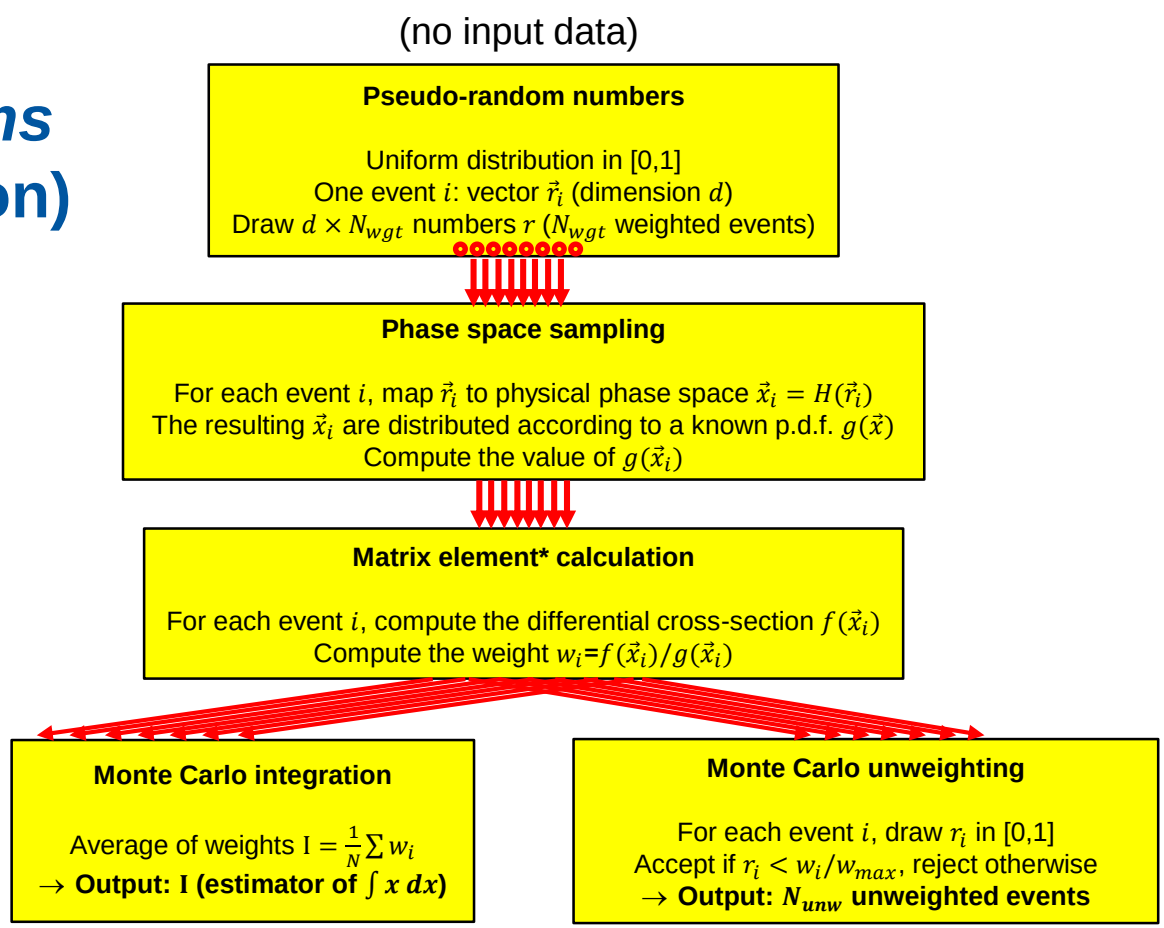
Generators lend themselves naturally to exploiting event-level parallelism via **data-parallel paradigms****

- **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 (where if/then branches are frequent and lead to thread divergence on GPUs)

Potential interest of GPUs
- Faster (cheaper?) than on CPUs
- Exploit GPU-based HPCs

WIP for MG5aMC on GPUs (planned WG talk) – see next slide



**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*

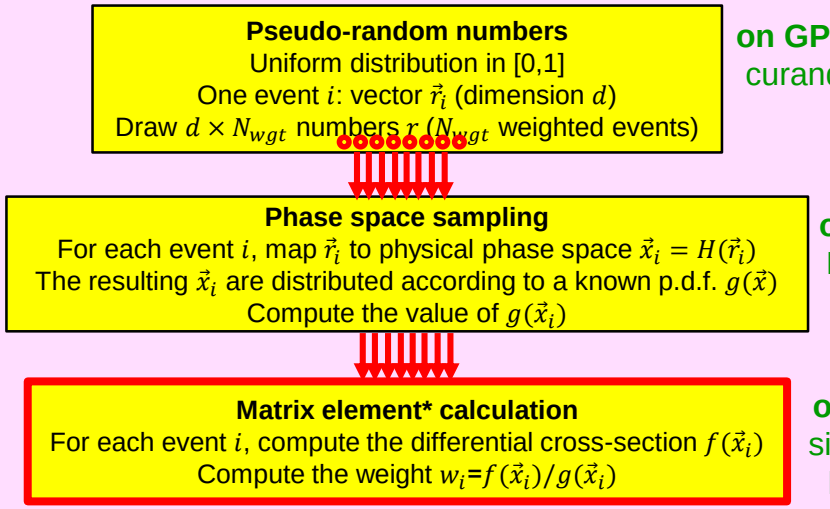
***This simple event-level parallelism can also be used as the basis for task-parallel approaches (multi-threading or multi-processing)*

WIP: MadGraph5 on GPU (Louvain, CERN, Argonne)

Why MadGraph?
- One of the main generators (especially in CMS)
- Expand on previous GPU work at KEK
- Work with main MG developer (O. Mattelaer)

WIP in several directions
- Modernize/Improve previous KEK work on BASES, SPRING, VEGAS, HEGET in CUDA
- Look into Alpaka and hipSYCL abstractions
- Port and optimize the current matrix-element code-generation engine (ALOHA, replacing HELAS/HEGET) in CUDA

A very nice collaboration of people with diverse skills!
- Theorists, experimentalists, software engineers
- Some of us benefitted enormously from the help of external mentors in the Sheffield GPUHackathon



on GPU curand

on GPU Rambo kernel

on GPU sigmakin kernel

Very preliminary status: *CUDA GPU throughput (Nvidia V100) is a factor ~200-1600 higher than CPP on a single CPU core*

Current proof-of-concept prototype:
- Focus on a simple process e+e- to mu+mu-, only 2 Feynman diagrams i.e. few lines of code
- Start from auto-generated C++, hardcode CUDA changes, test, optimize, move upstream to code generating code (python), try out on gg to ttgg
- No PDF yet, simple/inefficient sampler (Rambo)

Full chain on the GPU (random numbers, map to momenta, compute ME) – for this simple ME, now *bottleneck is copy from GPU to CPU* of momenta, weights and MEs (will be different e.g. in gg to ttgg)

No complete cross-section calculation yet, but it's easy to add to complete the proof-of-concept

NB: no evidence for "thread divergence"
All threads execute the same operation
Analogous to SIMD vectorization on a CPU

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 $\vec{x}_i = H(\vec{r}_i)$
The resulting $\vec{x}_i$ are distributed according to a known p.d.f. $g(\vec{x})$
Compute the value of $g(\vec{x}_i)$

Matrix element calculation

For each event i, compute the differential cross-section $f(\vec{x}_i)$
Compute the 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 sampling optimization

Compute w_{max} over the phase space
Optimize phase space sampling $H(\vec{r}_i)$:
EITHER minimize the variance of I
***OR* maximize the unweighting efficiency**

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 traditional techniques for sampling
- Importance, stratified, adaptive…
- Multi-channel

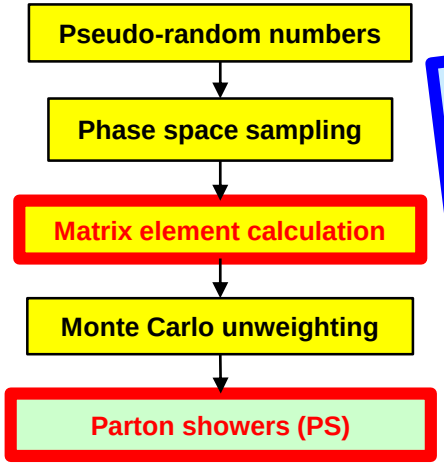
***New ML techniques** developed in many teams:*
- *BDT, DNN (WG talk by Bendavid, Jan 2020)*
- *Normalizing flows (2001.10028)*
- *Neural importance sampling (2001.05478)*
- *Dedicated future WG meeting?*

Example (normalizing flows): *Sherpa* W+jets @LO
- **Efficiency is ~30% for W+0jets (x2.2 better!)**
- **Efficiency is ~0.08% for W+4jets (x1.1 better!)**

Unweighting efficiency		LO QCD				
$\langle w \rangle / w_{max}$		$n=0$	$n=1$	$n=2$	$n=3$	$n=4$
$W^+ + n$ 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

Gao et al., https://arxiv.org/abs/2001.10028

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



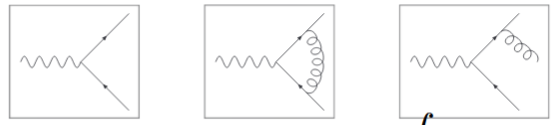
NB: ONLY THEORISTS CAN MAKE PROGRESS ON NLO MATCHING PRESCRIPTIONS!

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



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

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

S and H events: two separate sets of events (different matrix elements)

Integral = S+H is positive – but individual events can have negative weights

Work ongoing in many teams with different approaches:

- *Theory (NLO matching): MC@NLO-Δ (2002.12716)*
- *Theory (NLO matching): Sherpa (Danziger's thesis)*
- *ML (resampling): Positive Resampler (2005.09375)*
- *ML (resampling): Neural Resampler (2007.11586)*
- *Dedicated future WG meeting?*

For a fraction r of negative weight events: need a factor $\frac{1}{(1-2r)^2}$ more events to generate, simulate, reconstruct than for r=0

Example for $Hb\bar{b}$: r~40% implies ~25 times more events

MC@NLO-Δ may reduce some of these costs by a factor ~2

	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.

Frederix et al., https://arxiv.org/abs/2002.12716

Issue #5: *careers, funding, training and collaboration*

For instance: no obvious career recognition for theorists to motivate them to spend time on code speedups

Some say this is technical work, not academic research….

WG talk by A. Buckley July 2020 (Efficient Computing for HEP UK)

Real world problems

ECHEP Generators area summary
Andy Buckley, Marek Schoenherr

Incentives are often misaligned: need to design around this

- MC generators are organisationally developed under HEP theory grants
- Career progression still strongly coupled to "theory" papers, not sw performance
- Less institutional room & reward than in experiment for technical work
- If hiring fractional FTEs (we are!), how to ensure career path? Essential to attract right people, which is crucial.
- Organising how (remote?) fractional FTEs will embed in MC gen groups.
- If experimentalists, need their gen work FTE to earn service credit. Work on external tools which primarily benefit experiments needs to be allowed and recognised — perhaps by classifying as software, not physics

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.

No simple solution – a few directions:

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

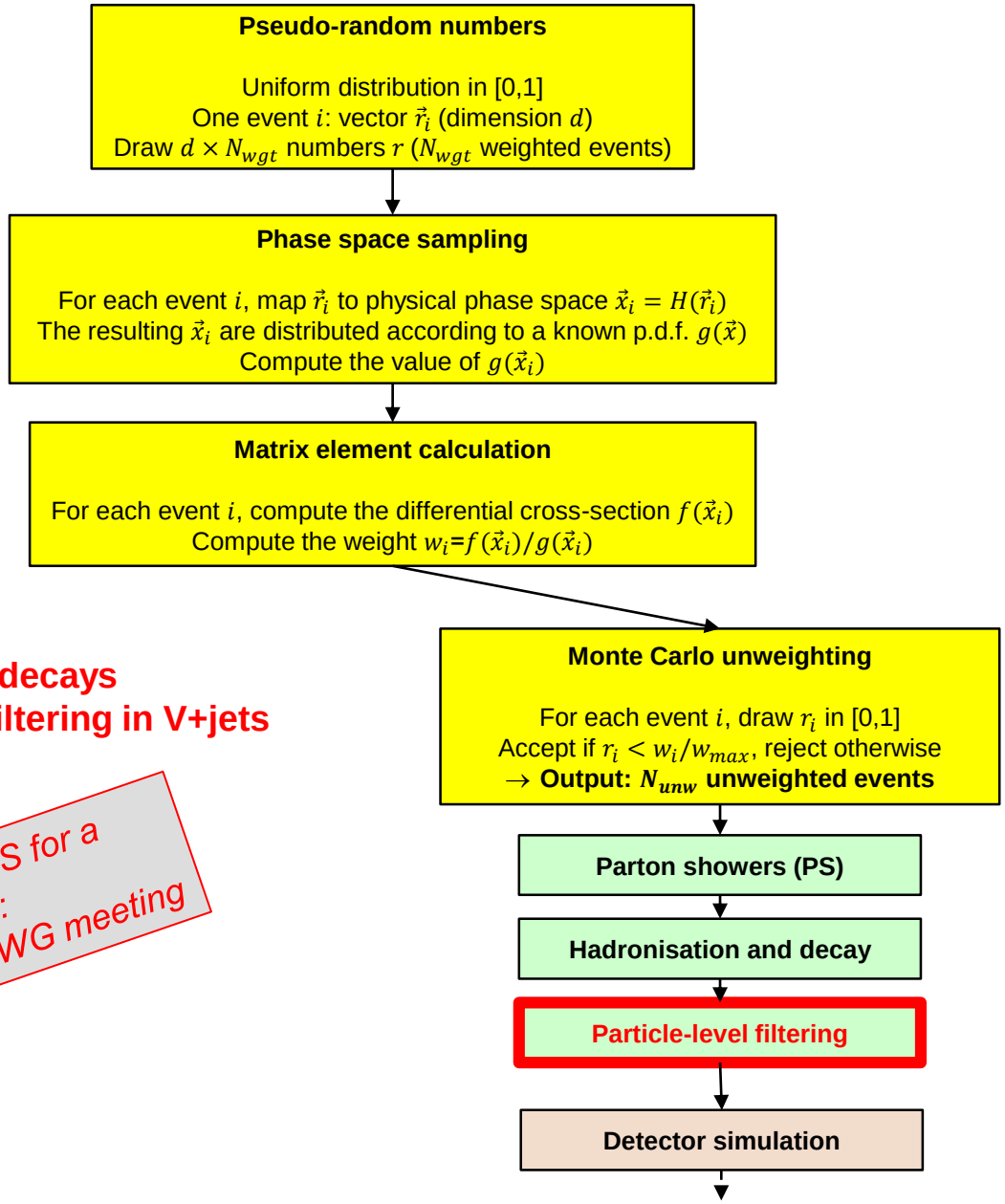
- ***Designing funding opportunities*** around this issue

- **Collaboration between physicists and software engineers** is essential and mutually beneficial (e.g. see MG5 on GPU); this also requires a shared terminology and set of concepts for more effective communication (one of the goals of our arxiv paper)

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

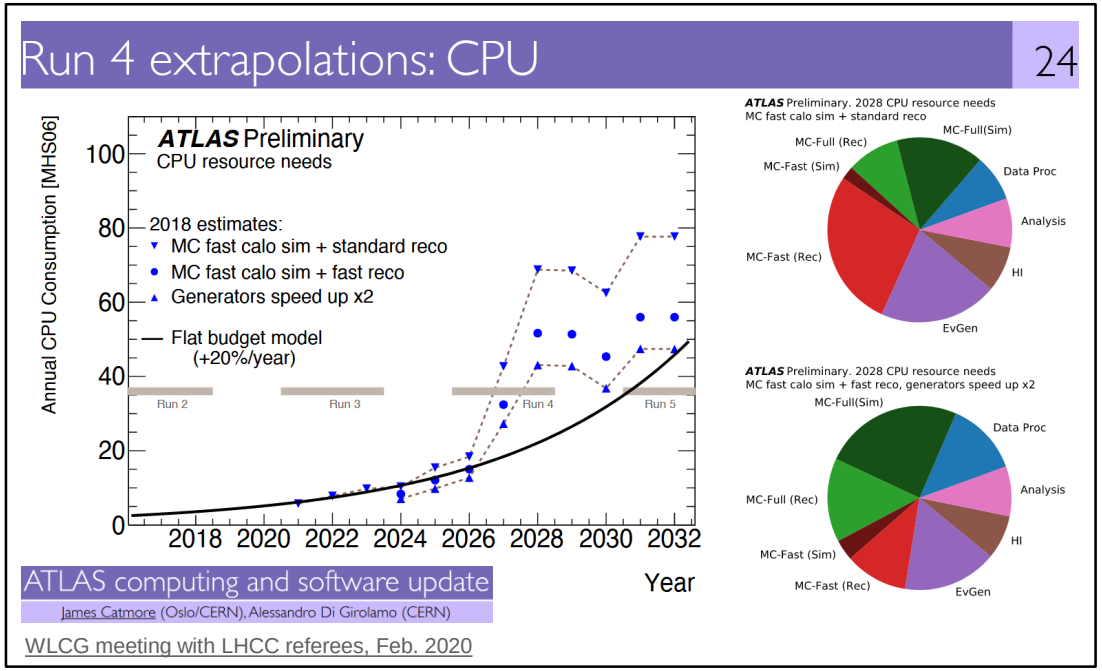


Examples
- **CMS: ~0.01% efficiency for specific Λ_B decays**
- **ATLAS: ~10% efficiency for B-hadron filtering in V+jets**

Interest in LHCb, CMS, ATLAS for a cross-experiment discussion:
- *Plan a dedicated future WG meeting*

Issue #7: *the future cost of improved precision*

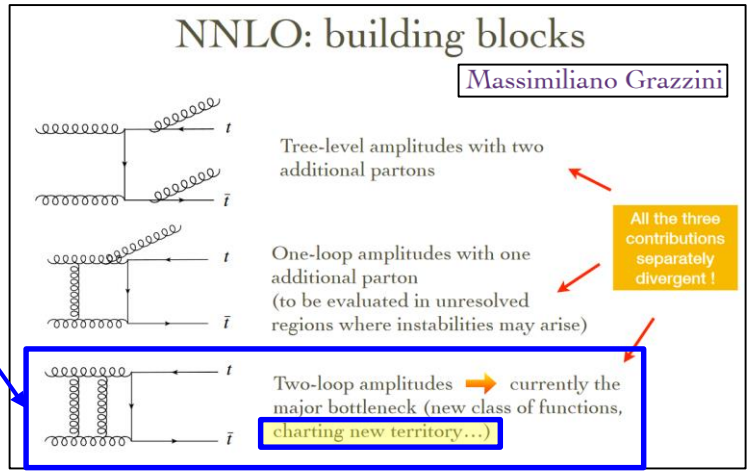
Work ongoing in many teams of theorists:
- WG talk by M. Grazzini July 2020
- Dedicated future meeting?



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

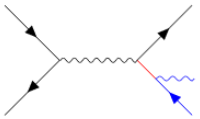
How much more would NNLO calculations cost?
- More Feynman diagrams (slower calculations)
- Two-loop diagrams (more complex, more expensive)
- More complex and inefficient phase space sampling?
- Higher fraction of negative weights in PS matching?

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



Issue #8: further speedups on CPUs are possible through technical or algorithmic improvements

Helicity Recylcing: Internal Wavefunction



- MadGraph recursively calculates internal wavefunctions
- For 8 helicity combinations red line has same value
- Aim of project: Only call internal wavefunction once and store it as number

Kiran Ostrolenk (UCL) | CPU performance improvements in MadGraph | My house., July 9, 2020 | 4 / 9

Example 1: helicity recycling in MadGraph

	$gg \to t\bar{t}$	$gg \to t\bar{t}g$	$gg \to t\bar{t}gg$	$qq \to t\bar{t}g$
Matrix reduction	55%	47%	31%	49%
Overall Speed-up	12%	23%	23%	11%

Table: $t\bar{t}$ production

Preliminary results

	$gg \to t\bar{t}$	$gg \to t\bar{t}g$	$gg \to t\bar{t}gg$
Matrix reduction	69%	70%	63%
Overall Speed-up		26%	50%

WG talk by K. Ostrolenk July 2020

Summary

- three configurations are bench-marked and optimized:

Configuration	Pythia8.244, sec	f_main2, sec	$\frac{\Delta t}{t_{Pythia8}}$, %
hardQCD+LHAPDF6	303.8	153.3	49.5
hardQCD+LHAGrid1	126.8	98.3	22.5
hardQCD+NNPDF	157.5	106.2	32.6

- All patches are applicable to previous Pythia8 versions as well as to Pythia83XX
- few patches are changing numerical results and more thorough validation is required.
- All patches are sent to Torbjörn Sjöstrand and will be included into next Pythia83X release.
- additional 10 % can be gained from modification of LHAPDF6 (hopefully will be included into next LHAPDF6 release).

Dmitri Konstantinov, Grigory LatyshevNR(| Pythia8.244 code optimization | 02.03.2020 | 21 / 24

Example 2: caching/recycling of PDF data in Pythia8

D. Konstantinov, EP-SFT talk Mar 2020 (planned to repeat as WG talk)

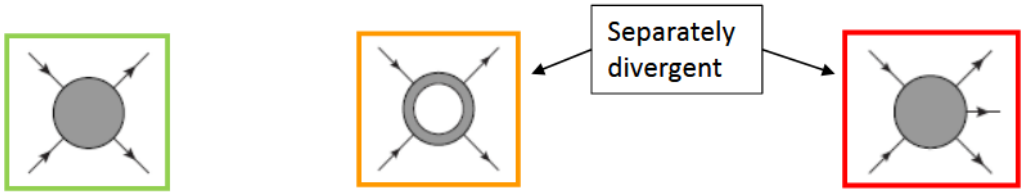
Conclusions: a multi-prong strategy for MC event generators

- Main WG priorities according to the conclusions of our paper: (not in order)
  - 1. Gain a better understanding of current CPU costs by accounting and benchmarking
  - 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 (also discussed in detail in the paper):
  - 6. Analyse filtering strategies and inefficiencies in the experiments
  - 7. Understand and estimate future additional costs due to NNLO and increased precision
  - 8. Other CPU speedup opportunities via software technicalities or algorithmic improvements
- **Each of these issues requires a different mix of skills and expertise**
  - E.g. research on negative weight needs theorists, a GPU port needs software experts
  - Cross-domain collaboration (with funding for all relevant profiles) is essential!

Backup slides

Structure of a NLO calculation

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



$$\langle O \rangle = \int O\, d\sigma = \int d\Phi_n\, O(\Phi_n)\, [B(\Phi_n) + V_b(\Phi_n)] + \int d\Phi_n\, d\Phi_r\, O(\Phi_n, \Phi_r)\, R(\Phi_n, \Phi_r)$$

- [Parametrized (n+1)-body phase space Φ_{n+1} in terms of Born and radiation $\Phi_{n+1} = \{\Phi_n, \Phi_r\}$]

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

$$\langle O \rangle = \int d\Phi_n\, O(\Phi_n) \left[B(\Phi_n) + V_b(\Phi_n) + \int d\Phi_r\, C(\Phi_n, \Phi_r) \right]$$
$$+ \int d\Phi_n\, d\Phi_r \underbrace{\left[O(\Phi_n, \Phi_r)\, R(\Phi_n, \Phi_r) - O(\Phi_n)\, C(\Phi_n, \Phi_r) \right]}_{\text{finite}}$$

- Defining: $V(\Phi_n) = V_b(\Phi_n) + \int d\Phi_r\, C(\Phi_n, \Phi_r) \quad \Longleftarrow$ finite

$$\langle O \rangle = \int d\Phi_n\, O(\Phi_n)\, [B(\Phi_n) + V(\Phi_n)] + \int d\Phi_n\, d\Phi_r\, [O(\Phi_n, \Phi_r)\, R(\Phi_n, \Phi_r) - O(\Phi_n)\, C(\Phi_n, \Phi_r)]$$

Hit and Miss

Hit and miss method:

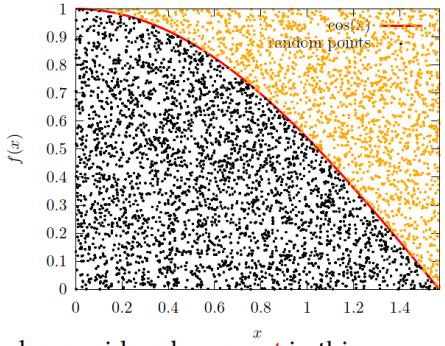
- throw N random points (x, y) into region.
- Count hits N_{hit}, i.e. whenever $y < f(x)$.

Then

$$I \approx V \frac{N_{\text{hit}}}{N}.$$

approaches 1 again in our example.

Example: $f(x) = \cos(x)$.



Every accepted value of x can be considered an event in this picture. As $f(x)$ is the 'histogram' of x, it seems obvious that the x values are distributed as $f(x)$ from this picture.

Importance sampling

Error on Crude MC $\sigma_{MC} = \sigma/\sqrt{N}$.
$\Longrightarrow$ Reduce error by reducing variance of integrand.

Idea: *Divide out the singular structure.*

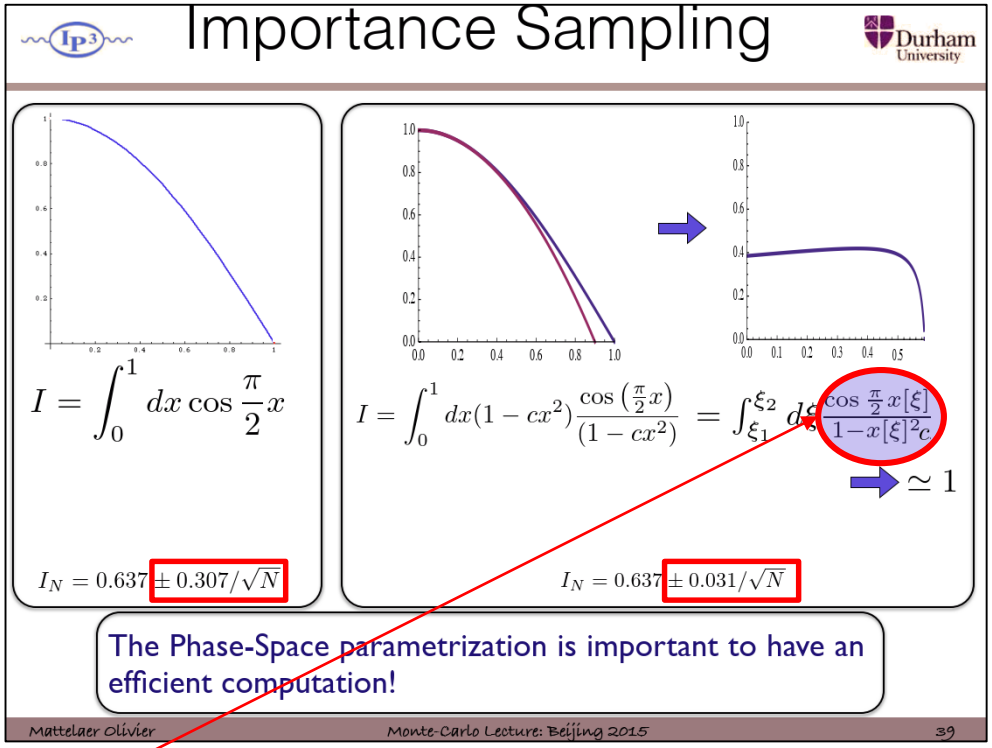
$$I = \int f \, \mathrm{d}V = \int \frac{f}{p} p \, \mathrm{d}V \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}.$$

where we have chosen $\int p \, \mathrm{d}V = 1$ for convenience.

Note: need to sample flat in $p \, \mathrm{d}V$, so we better know $\int p \mathrm{d}V$ and it's inverse.

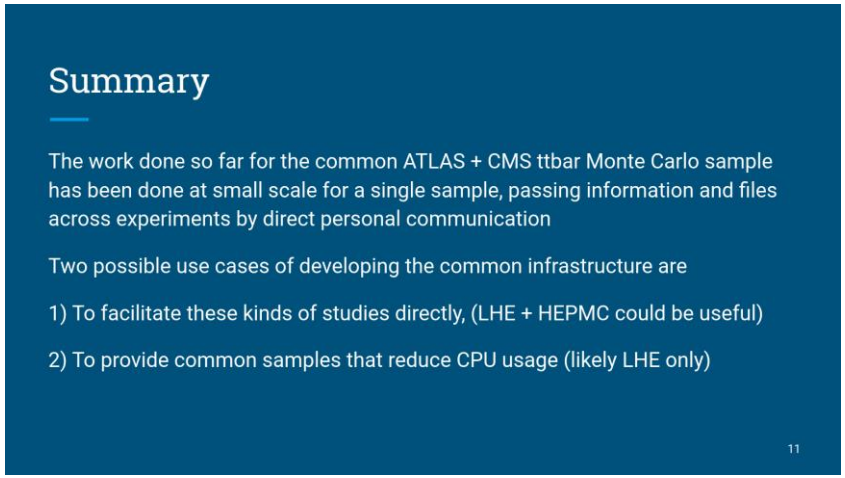
Choose p as close to f as possible.

Importance Sampling



https://cp3.irmp.ucl.ac.be/projects/madgraph/raw-attachment/wiki/FHEP

In addition: further cost savings are possible by reusing events



Example 1: sample sharing between ATLAS and CMS

A common EOS space has been created for the WG

WG talk June 2019 by K. Cormier and J. Fernandez Menendez

Example 2: extend the use of event reweighting