Efficient Phase Space Generation for Particle Reactions

PHS2350 Research Project By Zara Rosenberg
Supervisor: Prof. Peter Skands
Co-supervisor: Ludovic Scyboz

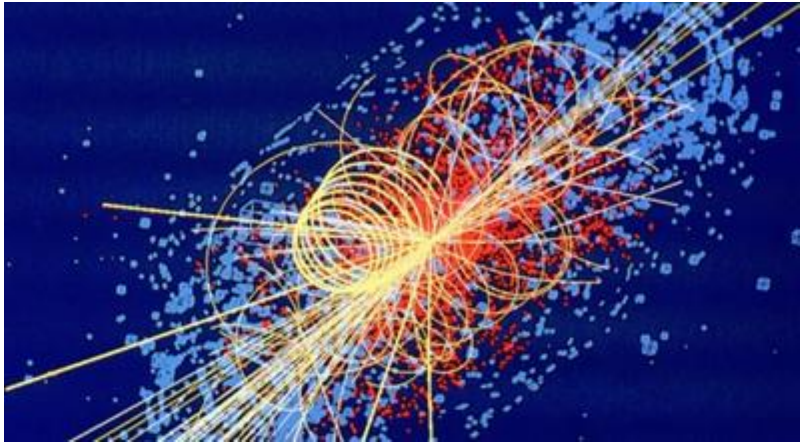
Introduction

Phase Space generators

- Computer simulations
- Predict experiments and test theories
- Heavily peaked probability distributions

Objective:

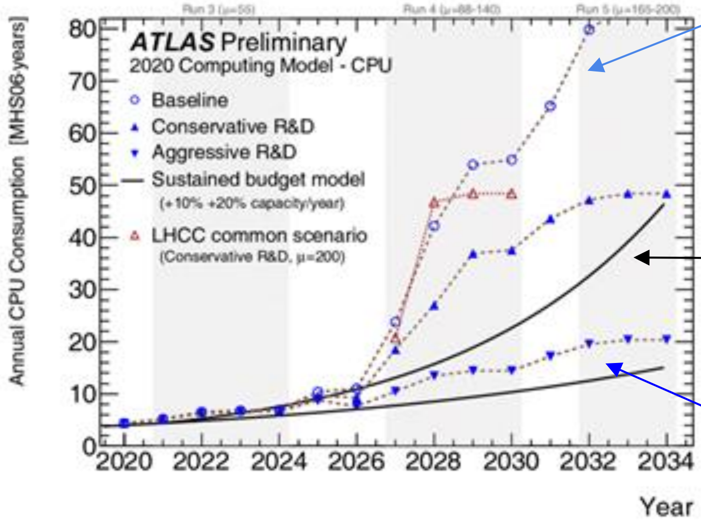
- Validate new method of phase space generation



Simulated data modelled for the CMS detector on the Large Hadron Collider (LHC) at CERN, L Taylor [http://cds.cern.ch/record/39444]

Significance

ATLAS computing crisis



Current CPU requirements

CPU capability

CPU requirements with aggressive R&D

[ATLAS Computing Crisis: Marcon, et al., The European Physical Journal Conferences (2021)]

Environmental impact

Convenience

Background

Probability density

Differential phase space element

$$\underbrace{P(\Phi_n)}\underbrace{d\Phi_n}$$

$$d\Phi_n = \prod_{i=1}^{n} \frac{dp_{ix}dp_{iy}dp_{iz}}{2E_i(2\pi)^3}$$

See, e.g., PDG, PRD110 (2024) 3

Background

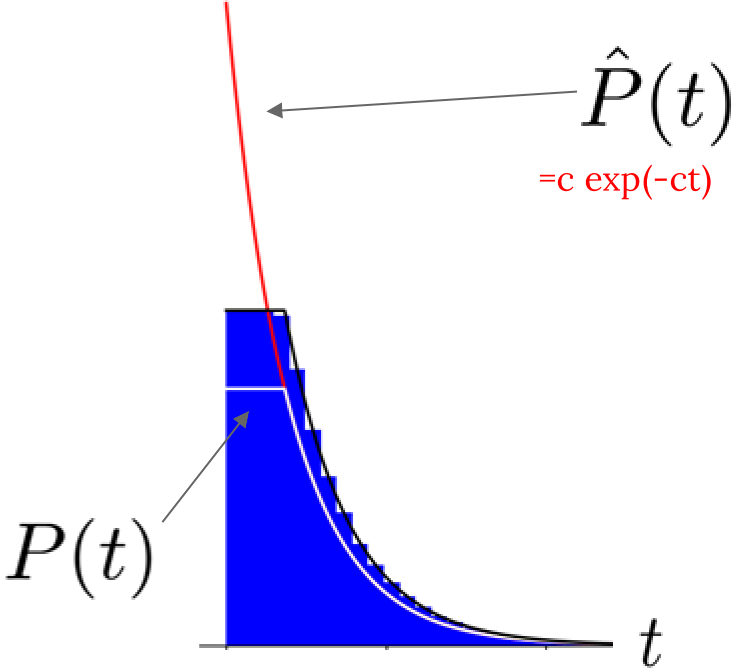
Acceptance probability

Simpler probability density that overestimates P

$$\overbrace{\frac{P(\Phi_n)}{\hat{P}(\Phi_n)}}^{\text{Acceptance probability}} \underbrace{\hat{P}(\Phi_n)d\Phi_n}_{\text{SUNSHINE phase space generator}}$$

SUNSHINE phase space generator

The Veto Algorithm

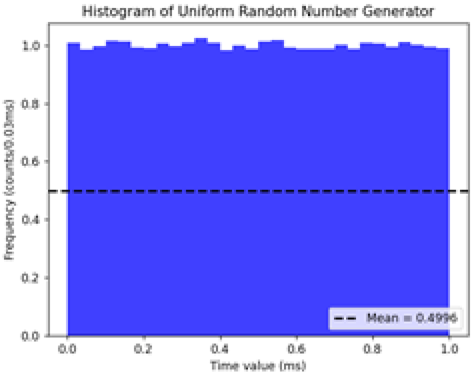


1. Choose a simple overestimating function
2. Use ITM to generate a T
3. Generate r
4. $r \leq \frac{P(T)}{\hat{P}(T)}$
5. Accept/reject

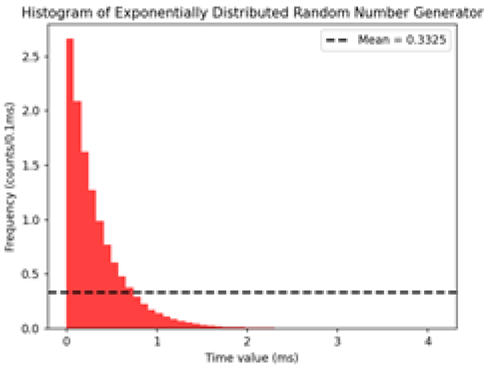
Veto algorithm: See, e.g, Mrenna, Skands, Phys.Rev.D 94. 2016

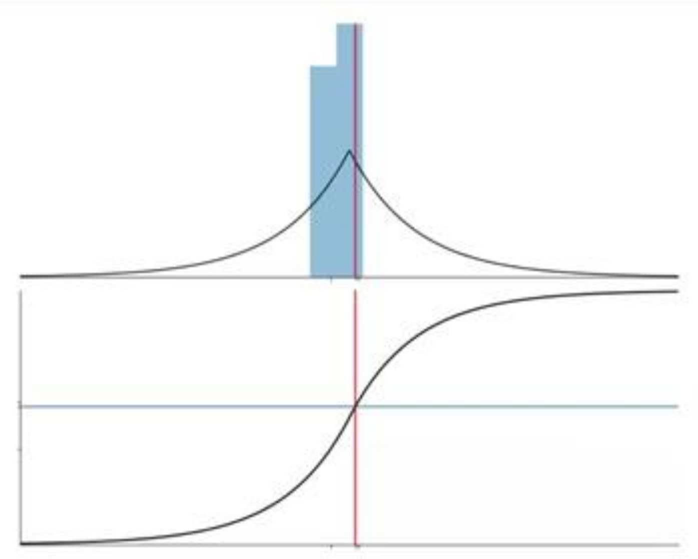
Inverse Transform Method

Uniform Distribution



Inverse Transform Method (ITM) for an Exponential Distribution





numPy: Harris, et al. *Nature 585 (2020)* 357

Inverse Transform Method: Press, W, et al. (1994). Numerical Recipes in FORTRAN, The Art of Scientific Computing, Second Edition. New York: Cambridge University Press

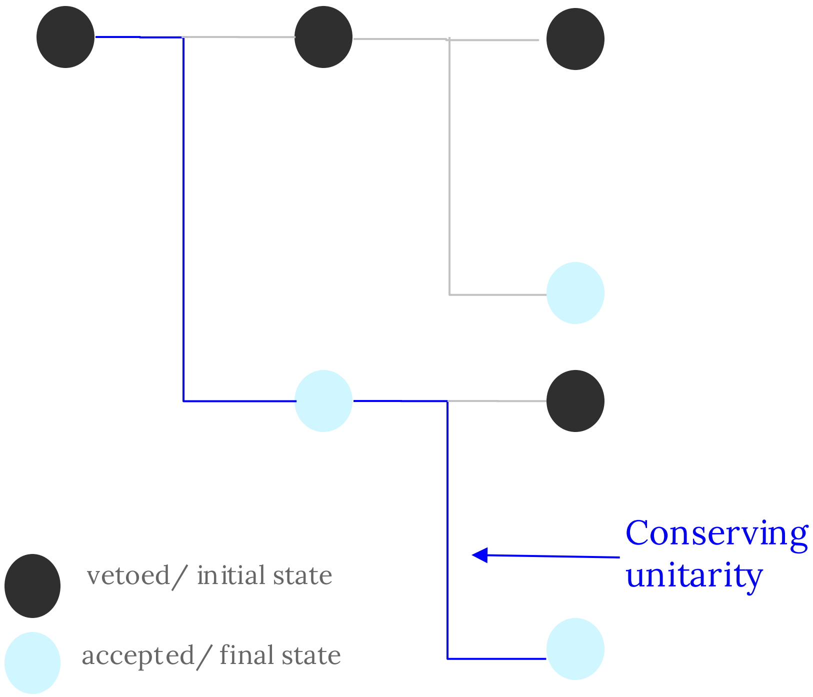
Animation: Vieira. T, (2020, June 30). Animation of the inverse transform method. Graduate Descent

SUNSHINE

- New approach to phase space generation
- Uses a modified veto algorithm
- No initialisation of antenna functions required

Sunshine background Stock Photos, Royalty Free Sunshine background Images | Depositphotos

Violating Unitarity



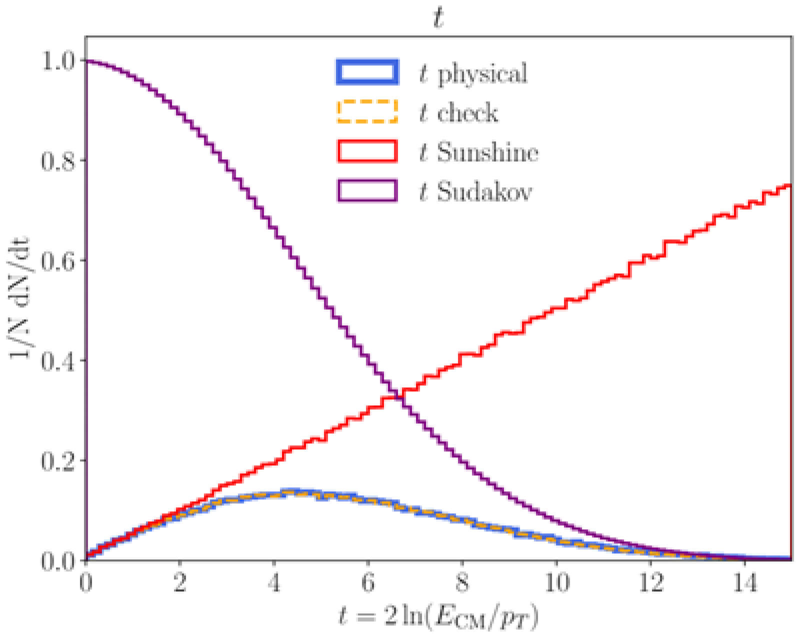
Application and Validation of Phase Space Generator

Antenna function

Sudakov factor (Δ)

$$c(t)exp(-\int_0^t c(t')dt')$$

Physical



PYTHIA: Bierlich et al., SciPost Phys.Codeb. 2022

Limitations

- Does not account for all peaked structures
- Only handles ordered structures
- Limited to the accuracy at which the antenna function is known

Conclusion

- SUNSHINE approach has been validated
- Efficient phase space generator
- Help solve some current problems we face

Proof

Unmodified veto algorithm: $\frac{d\mathcal{P}}{dt} = P(t)\Delta(t_0, t)$

Sudakov Factor: $\Delta_i(t_i, t_{i+1}) = exp(-\int_{t_{i+1}}^{t_i} ant_{i\to i+1}(\tilde{t}_{i+1})d\tilde{t}_i)$

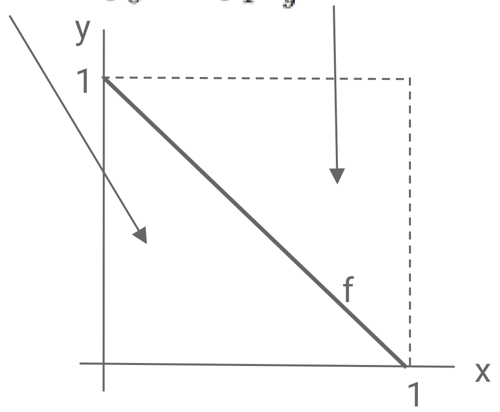
Arbitrary path of emission/decay:

$$\frac{d\mathcal{P}_{m_1,m_2,\ldots,m_n}}{dt} = \prod_{j=1}^{m_1}\int_{t_1}^{\tilde{t}_{m_1-1}} ant_{0\to 1}(\tilde{t}_j)d\tilde{t}_j \prod_{k=1}^{m_2}\int_{t_2}^{\tilde{t}_{m_2-1}} ant_{1\to 2}(\tilde{t}_k)d\tilde{t}_k \ldots \prod_{l=1}^{m_n}\int_{t_n}^{\tilde{t}_{m_n-1}} ant_{n-1\to n}(\tilde{t}_l)d\tilde{t}_l \prod_{i=0}^{n-1} ant_{i\to i+1}\Delta_i(t_i, t_{i+1})$$

Hypertriangle Identity

$$\int_0^1 f(x)dx \int_0^x f(y)dy = \frac{1}{2}\left(\int_0^1 dx \int_0^x dy f(x)f(y) + \underbrace{\int_0^1 dx \int_0^x dy f(x)f(y)}\right)$$

$$\int_0^1 dx \int_0^x dy f(x)f(y)) = \int_0^1 dy \int_{1-y}^1 dx f(x)f(y)$$



$$= \frac{1}{2}\left(\int_0^1 f(x)dx \int_0^1 dy f(y)\right)$$

$$= \frac{1}{2!}\left(\int_0^1 f(x)dx\right)^2$$

$$\int_0^1 f(x)dx \int_0^x f(y)dy = \frac{1}{2!}(\int_0^1 f(x)dx)^2 \qquad \Delta_i(t_i, t_{i+1}) = exp(-\int_{t_{i+1}}^{t_i} ant_{i\to i+1}(\tilde{t}_{i+1})d\tilde{t}_i)$$

Proof

$$\frac{d\mathcal{P}}{dt} = \underbrace{\sum_{m_1=0}^{\infty} \frac{1}{m_1!}(\int_{t_1}^{t_0} ant_{0\to 1}(\tilde{t}_j)d\tilde{t}_j)^{m_1}}_{= exp(\int_{t_1}^{t_0} ant_{0\to 1}(\tilde{t}_j)d\tilde{t}_j) = \frac{1}{\Delta_0}} \dots \sum_{m_n=0}^{\infty} \frac{1}{m_n!}(\int_{t_n}^{t_{n-1}} ant_{n-1\to n}(\tilde{t}_l)d\tilde{t}_l)^{m_n} \prod_{i=0}^{n-1} ant_{i\to i+1}\Delta_i(t_i, t_{i+1})$$

$$\frac{d\mathcal{P}}{dt} = \frac{1}{\Delta_0} \dots \frac{1}{\Delta_{n-1}} \prod_{i=0}^{n-1} ant_{i\to i+1}(t_{i+1}) \Delta_0 \dots \Delta_{n-1}$$

$$\frac{d\mathcal{P}}{dt} = \prod_{i=0}^{n-1} ant_{i\to i+1}(t_{i+1})$$