What is inside MC generators...
...and why it is wrong

Tomasz Golan
University of Rochester / Fermilab

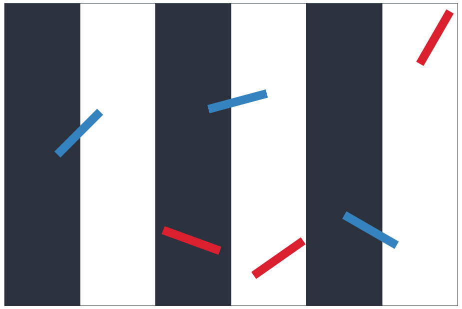
NuSTEC, Okayama 2015

Monte Carlo method

Buffon's needle problem

Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?

Georges-Louis Leclerc,
Comte de Buffon
18th century



blue are good

red are bad

Monte Carlo without computers

If needle length (l) $<$ lines width (t):

$$P = \frac{2l}{t\pi}$$

which can be used to estimate π:

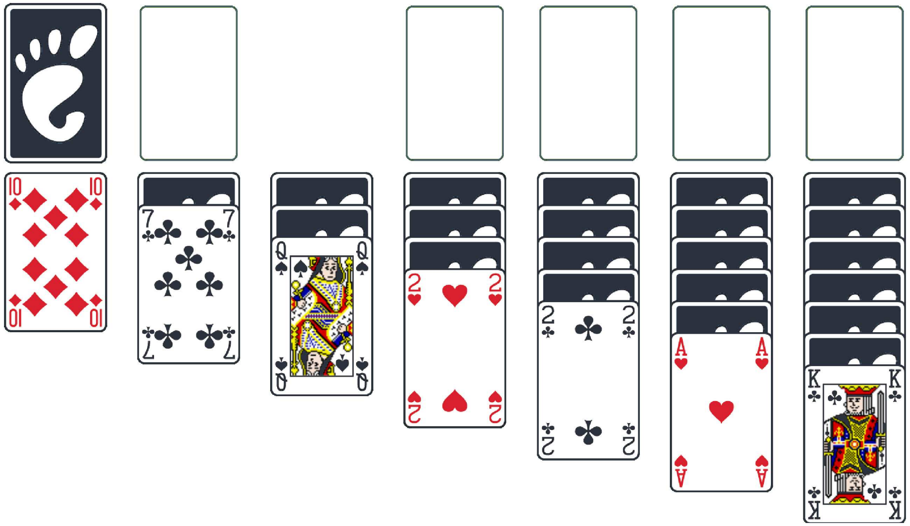
$$\pi = \frac{2l}{tP}$$

MC experiment was performed by Mario Lazzarini in 1901 by throwing 3408 needles:

$$\pi = \frac{2l \cdot 3408}{t \cdot \#red} = \frac{355}{113} = 3.14159292$$

From Solitaire to Monte Carlo method

- Stanisław Ulam was a Polish mathematician
- He invented the Monte Carlo method while playing solitaire
- The method was used in Los Alamos, performed by ENIAC computer



- What is a probability of success in solitaire?
  - Too complex for an analytical calculations
  - Lets try $N = 100$ times and count wins
  - With $N \to \infty$ we are getting closer to correct result

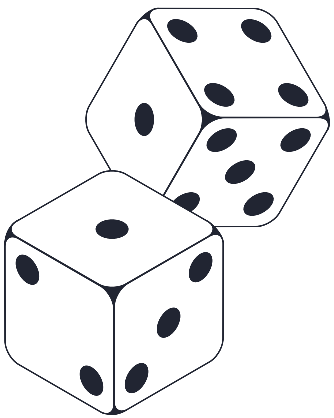
Newton-Pepys problem

Which of the following three propositions has the greatest chance of success?

A *Six fair dice are tossed independently and at least one "6" appears.*

B *Twelve fair dice are tossed independently and at least two "6"s appear.*

C *Eighteen fair dice are tossed independently and at least three "6"s appear.*



- First, lets go back to high school and calculate this analytically
- Let $p = \frac{1}{6}$ be the probability of rolling 6
- The probability of not rolling 6 is $(1 - p)$

A six attempts, at least one six

$$P_A = 1 - (1-p)^6 \approx 0.6651$$

B twelve attempts, at least two sixes

$$P_B = 1 - (1-p)^{12} - \binom{12}{1} p(1-p)^{11} \approx 0.6187$$

C eighteen attempts, at least three sixes

$$P_C = 1 - (1-p)^{18} - \binom{18}{1} p(1-p)^{17} - \binom{18}{2} p^2(1-p)^{16} \approx 0.5973$$

Newton-Pepys problem: MC attempt

- MC attempt is just “performing the experiment”, so we will be rolling dices
- Roll $6n$ times and check if number of sixes is greater or equal n
- Repeat N times and your probability is given by:

$$P = \frac{\text{number of successes}}{N}$$

```
def throw (nSixes):
  n = 0
  for _ in range (6 * nSixes):
    if random.randint (1, 6) == 6: n += 1
  return n >= nSixes

def MC (nSixes, nAttempts):
  n = 0
  for _ in range (nAttempts):
    n += throw (nSixes)
  return float (n) / nAttempts

if __name__ == "__main__":
  for i in range (1, 4):
    print MC (i, 1000)
```

Newton-Pepys problem: summary

- Your MC result depends on N
- Results for $N = 100$:

$$P_A = 0.71, 0.68, 0.76, 0.65, 0.68 \qquad P_A^{true} = 0.6651$$
$$P_B = 0.70, 0.56, 0.60, 0.63, 0.69 \qquad P_B^{true} = 0.6187$$
$$P_C = 0.62, 0.62, 0.53, 0.57, 0.62 \qquad P_C^{true} = 0.5973$$

- Results for $N = 10^6$:

$$P_A = 0.6655, 0.6648, 0.6653, 0.6662, 0.6653$$
$$P_B = 0.6188, 0.6191, 0.6191, 0.6190, 0.6182$$
$$P_C = 0.5975, 0.5979, 0.5972, 0.5978, 0.5973$$

- Your MC results also depends on the way how random numbers were generated

Pseudorandom number generator

- PRNG is an algorithm for generating a sequence of "random" numbers
- Example: middle-square method (used in ENIAC)
  - take n-digit number as your seed
  - square it to get $2n$-digit number (add leading zeroes if necessary)
  - n middle digits are the result and the seed for next number
- Middle-square method for $n = 4$ and base seed $= 1111$:

$$\begin{aligned} 1111^2 &= 01234321 \to 2343 \\ 2343^2 &= 05489649 \to 4896 \\ &\vdots \\ 1111^2 &= 01234321 \to 2343 \end{aligned}$$

- Nowadays, more sophisticated PRNGs exist, but they also suffer on some common problems:
  - periodicity / different periodicity for different base seed
  - nonuniformity of number distributions
  - correlation of successive numbers

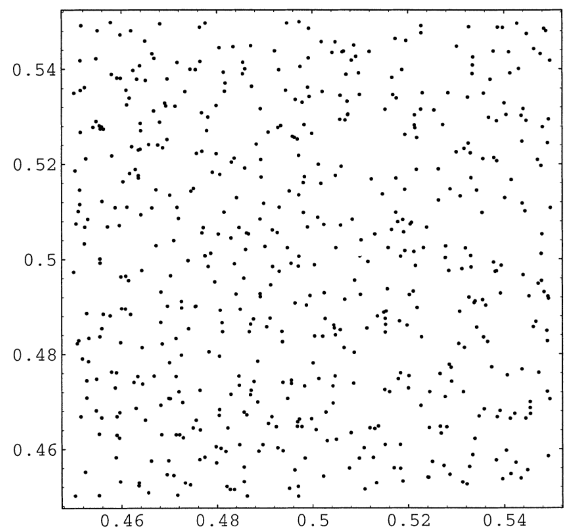


Fig. 1. LCG(2^{31}, 65539, 0, 1) Dimension 2: Zoom into the unit interval.

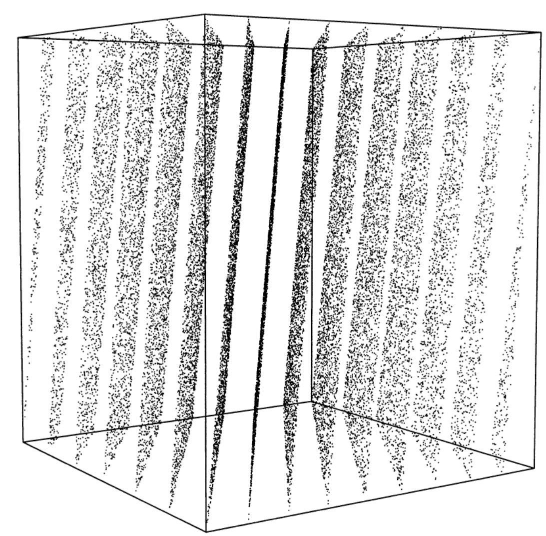


Fig. 2. LCG(2^{31}, 65539, 0, 1) Dimension 3: The 15 planes.

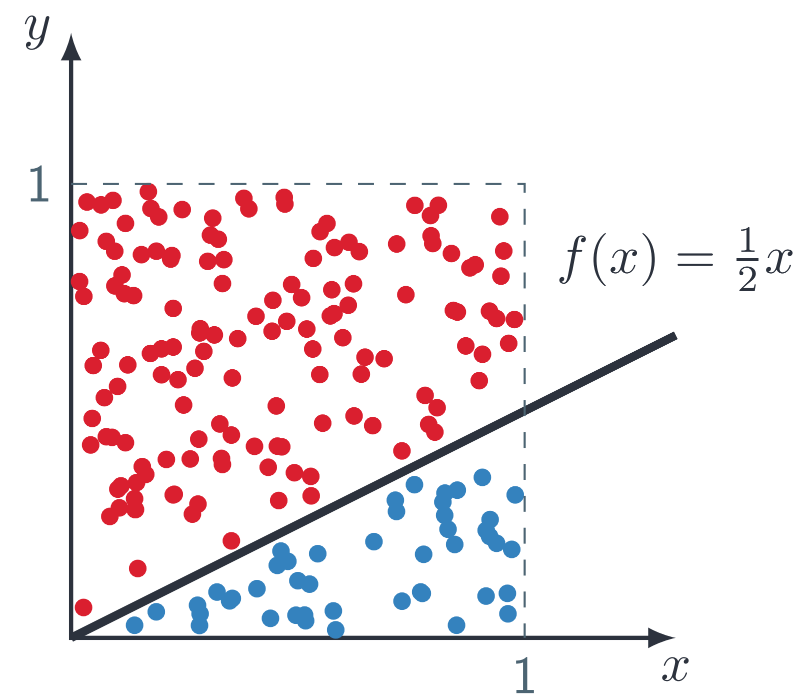
Mathematics and Computers in Simulations 46 (1998) 485-505

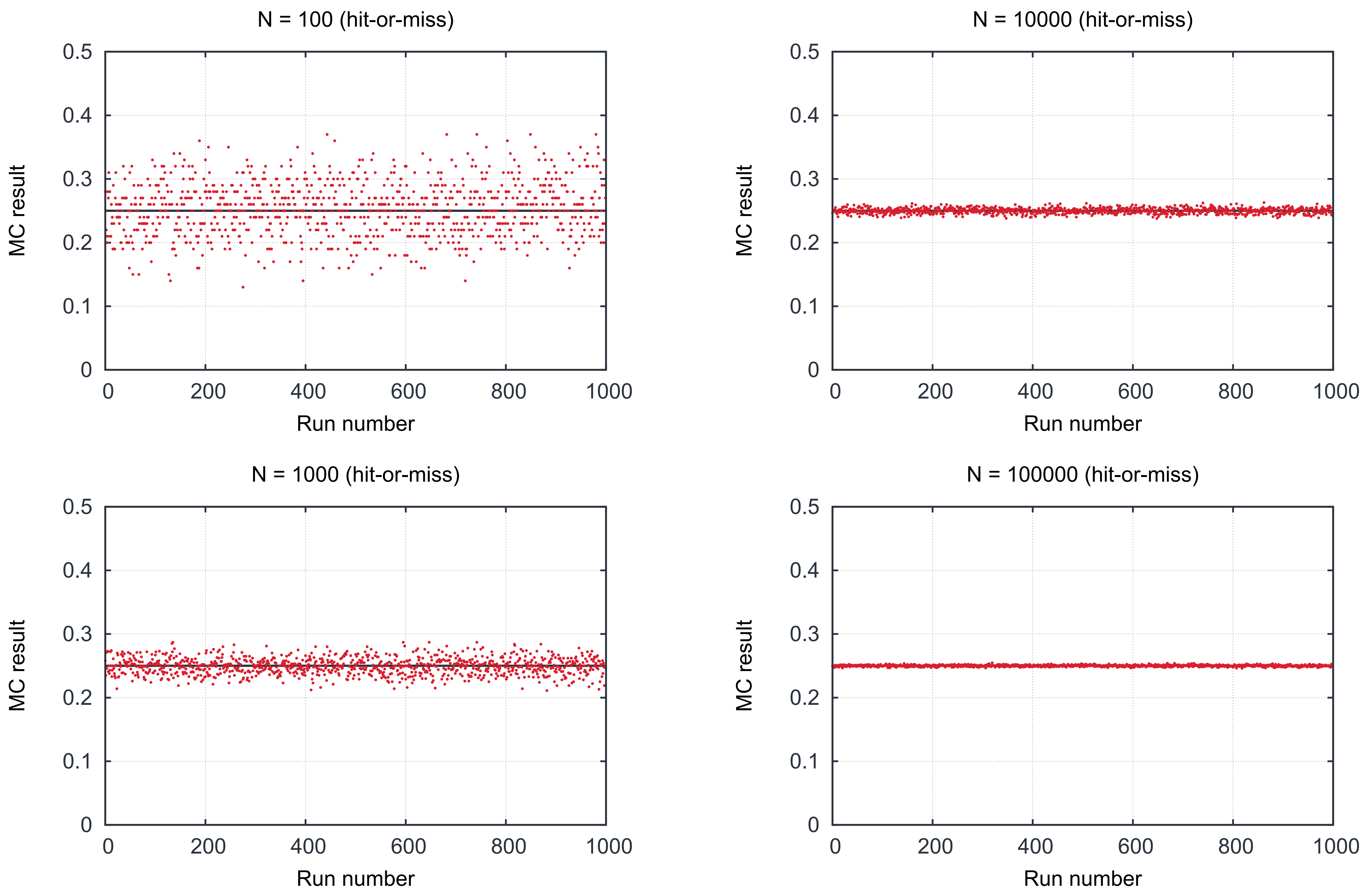
Lets do the following integration using MC method:

$$\int_0^1 f(x)dx = \int_0^1 \left(\frac{1}{2}x\right)dx = \frac{1}{2}\frac{x^2}{2}\Big|_0^1 = \frac{1}{4}$$

- take a random point from the $[0,1] \times [0,1]$ square
- compare it to your $f(x)$
- repeat N times
- count n points below the function
- you results is given by

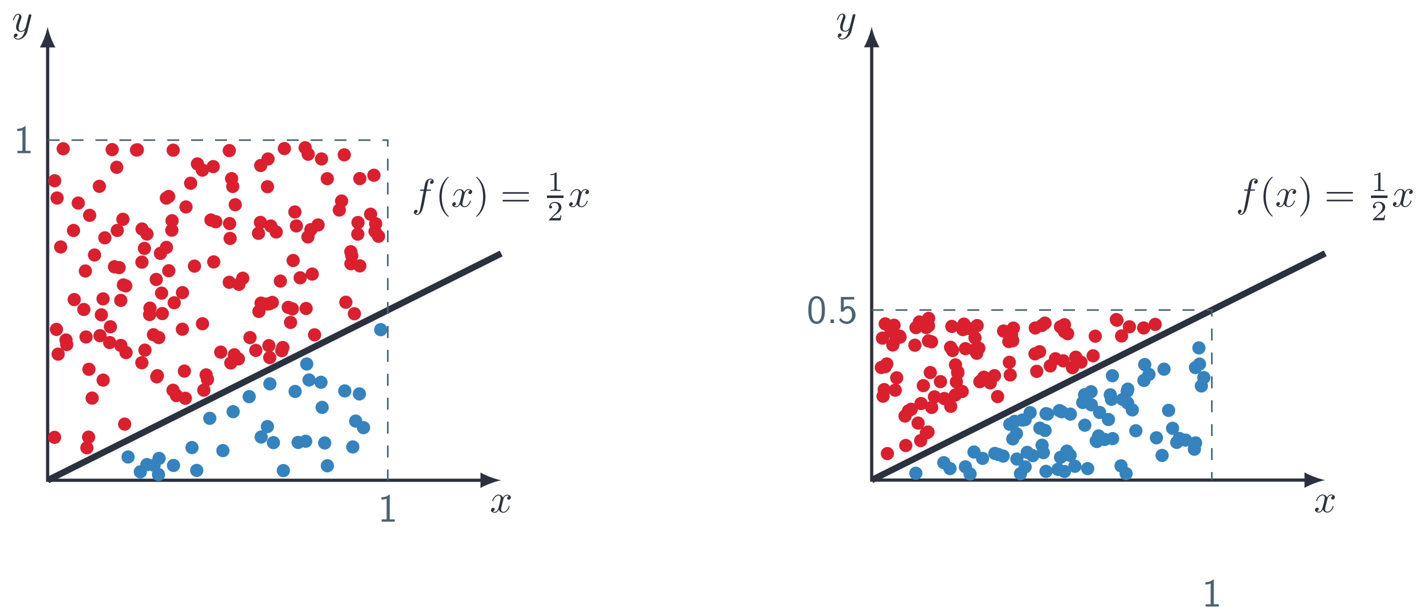
$$\int_0^1 f(x)dx = P_\square \cdot \frac{n}{N} = \frac{n}{N}$$





N = 100 (hit-or-miss)
N = 10000 (hit-or-miss)
N = 1000 (hit-or-miss)
N = 100000 (hit-or-miss)
MC result
Run number
0
0.1
0.2
0.3
0.4
0.5
0
200
400
600
800
1000

Optimization of MC

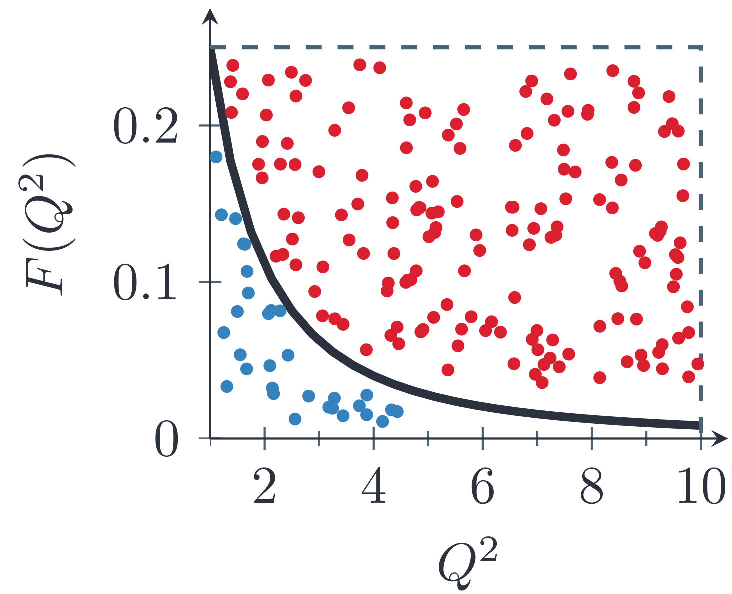


- You want to avoid generating "red" points as they do not contribute to your integral
- You can choose any rectangle as far as it contains maximum of $f(x)$ in given range

- Lets consider the following function:

$$F(Q^2) = \frac{1}{(1+Q^2)^2}$$

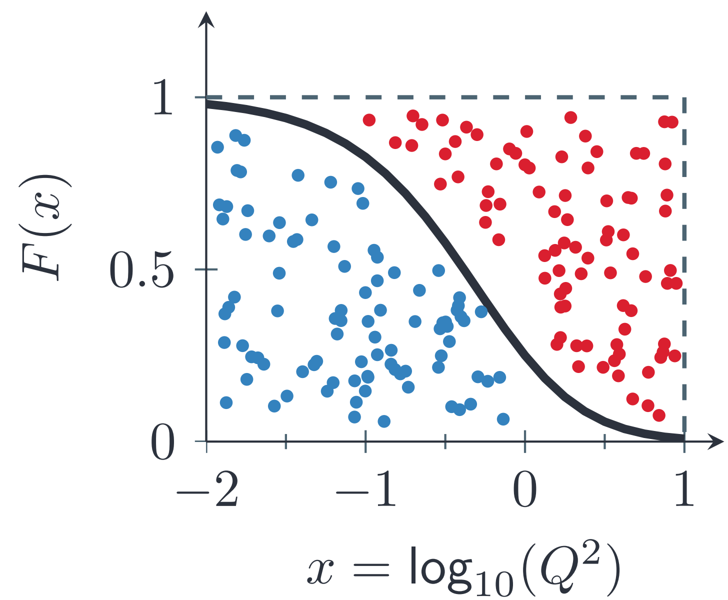
more or less dipole form factor



- Integrating this function over Q^2 is highly inefficient
- However, one can integrate by substitution to get better performance, e.g.

$$x = \log_{10}(Q^2)$$

don't forget about Jacobian



MC integration (crude method)

Lets do the following integration using MC method once again:

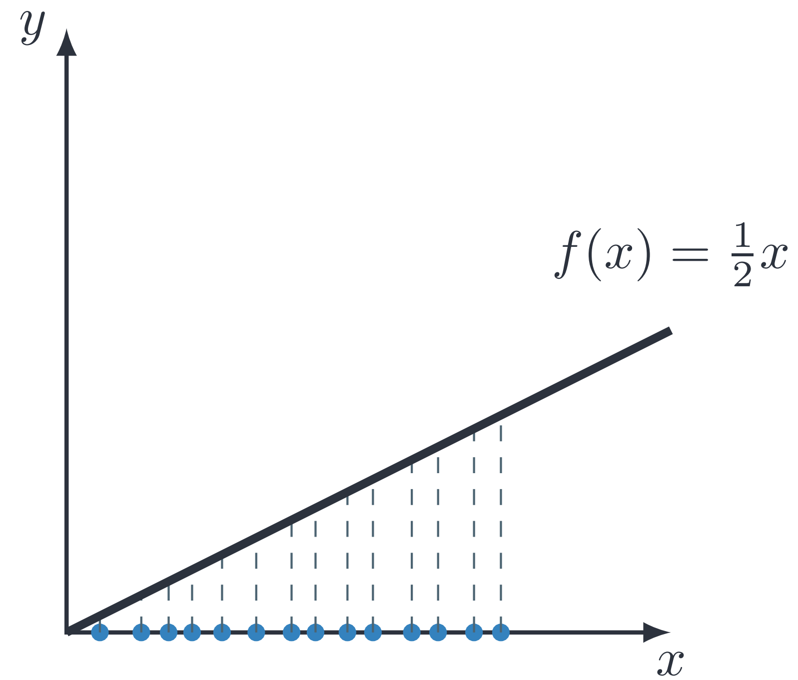
$$\int_0^1 f(x)dx = \int_0^1 \left(\frac{1}{2}x\right) dx = \frac{1}{2}\frac{x^2}{2}\Big|_0^1 = \frac{1}{4}$$

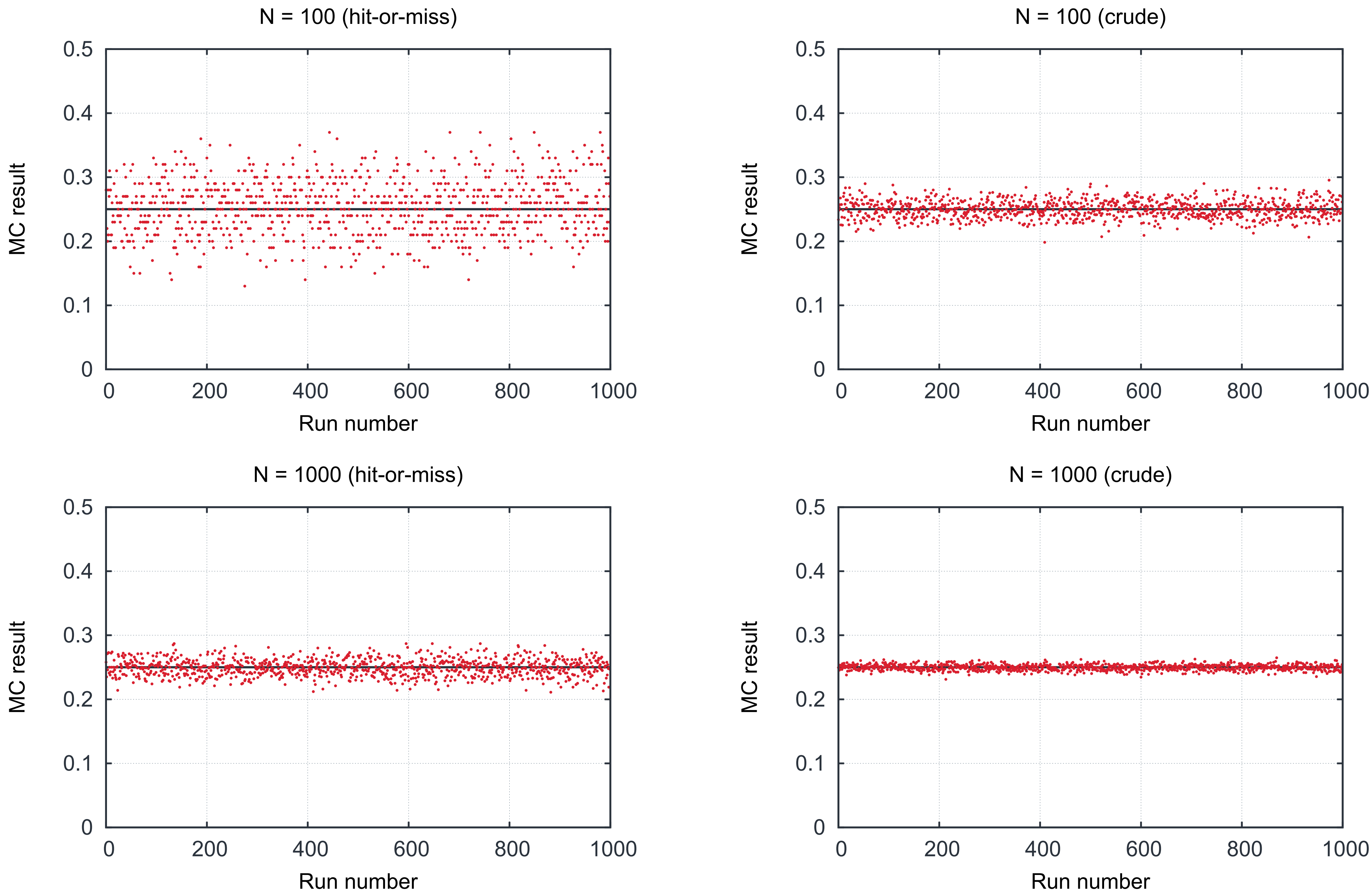
- One can approximate integral

$$\int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

where x_i is a random number from $[a, b]$

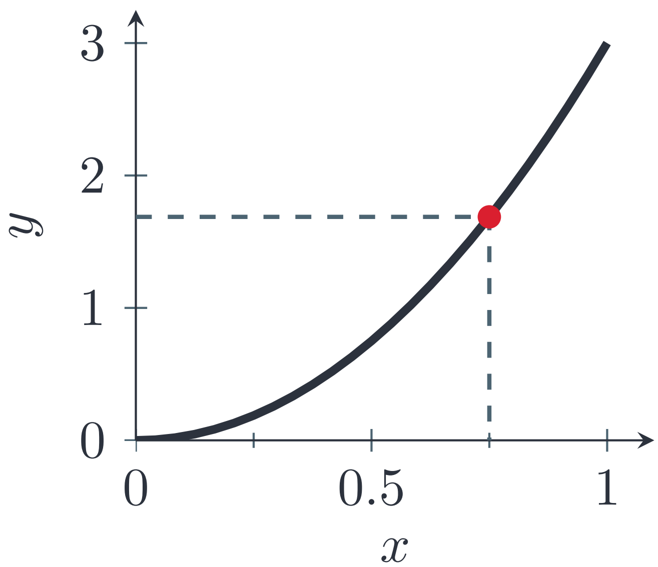
- It can be shown that crude method is more accurate than hit-or-miss
- We will skip the math and look at some comparisons





N = 100 (hit-or-miss)
N = 100 (crude)
N = 1000 (hit-or-miss)
N = 1000 (crude)
MC result
Run number
0
0.1
0.2
0.3
0.4
0.5
200
400
600
800
1000

- How to generate a random number from probability density function?
- Lets consider $f(x) = 3x^2$
- Which means that $x = 1$ should be thrown 2 times more often than $x = \frac{\sqrt{2}}{2}$



Cumulative distribution function

- Cumulative distribution function of a random variable X:

$$F(x) = P(X \leq x)$$

Note: $0 \leq F(x) \leq 1$ *for all* x

- Discrete random variable X:

$$F(x) = \sum_{x_i \leq x} f(x_i)$$

where f is probability mass function (PMF)

- Continuous random variable X:

$$F(x) = \int_{-\infty}^{x} f(t) dt$$

where f is probability density function (PDF)

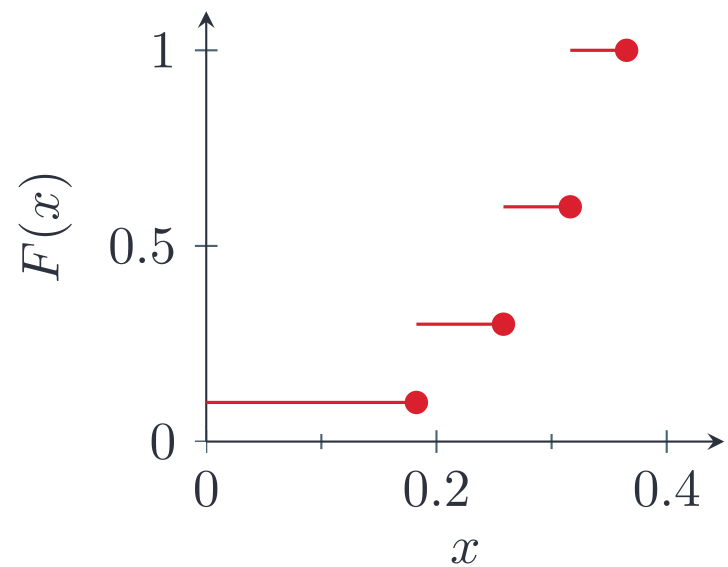
Cumulative distribution function - discrete example

- Probability mass function $f(x) = 3x^2$

  with discrete random variables X is $\{\sqrt{\frac{1}{30}}, \sqrt{\frac{2}{30}}, \sqrt{\frac{3}{30}}, \sqrt{\frac{4}{30}}, \}$

- CDF is given by:

$$F(x) = \begin{cases} \frac{1}{10} & \text{if} \quad x \le \sqrt{\frac{1}{30}} \\ \frac{3}{10} & \text{if} \quad x \le \sqrt{\frac{2}{30}} \\ \frac{6}{10} & \text{if} \quad x \le \sqrt{\frac{3}{30}} \\ \frac{10}{10} & \text{if} \quad x \le \sqrt{\frac{4}{30}} \end{cases}$$



- With $P = 1$ the random number is less or equal to $\sqrt{\frac{4}{30}}$, with $P = 0.6$ the random number is less or equal $\sqrt{\frac{3}{30}}$ ...

- To generate a random number from X according to $3x^2$:
  - generate a random number u from $[0, 1]$
  - if $u \leq 0.1$: $x = \sqrt{\frac{1}{30}}$
  - else if $u \leq 0.3$: $x = \sqrt{\frac{2}{30}}$ ...
- Results for $N = 10000$:

x	n	n/N	$f(x)$
$\sqrt{\frac{1}{30}}$	989	0.0989	0.1
$\sqrt{\frac{2}{30}}$	1959	0.1959	0.2
$\sqrt{\frac{3}{30}}$	2949	0.2949	0.3
$\sqrt{\frac{4}{30}}$	4103	0.4103	0.4

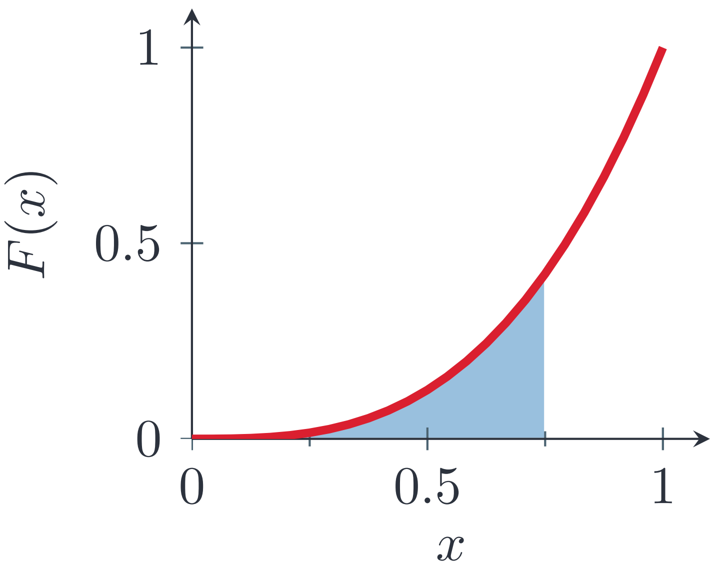
Cumulative distribution function - continuous example

- Probability density function $f(x) = 3x^2$

  with continuous random variables X range $[0,1]$

- CDF is given by:

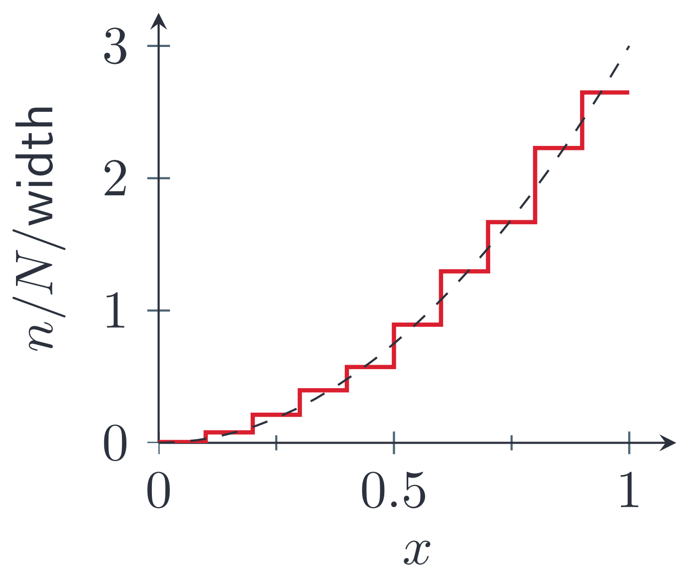
$$F(x) = \int_0^x f(t)dt = \int_0^x 3t^2 dt = t^3\Big|_0^x = x^3$$



- Blue area gives the probability that $x \leq 0.75$

Cumulative distribution function - continuous example

- To generate a random number from X according to $3x^2$:
  - generate a random number u from $[0, 1]$
  - find x for which $F(x) = u$, i.e. $x = F^{-1}(u)$
  - x is your guy
- Results for $N = 10000$:



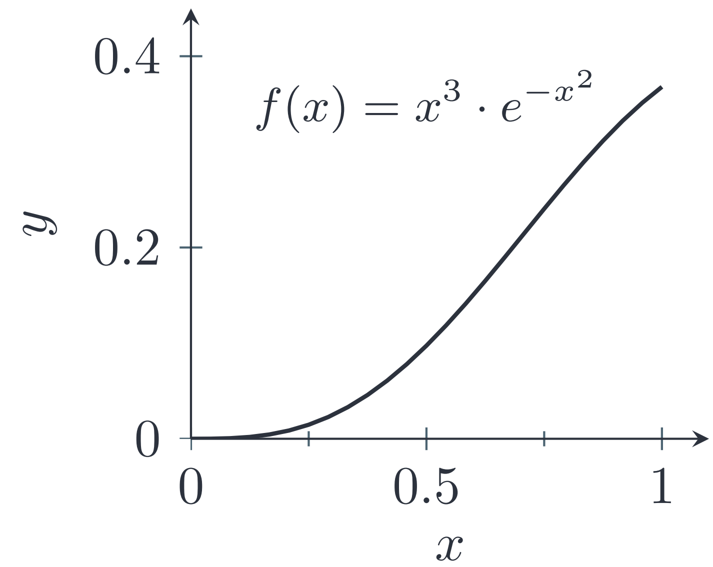
Unfortunately, usually F^{-1} is unknown, which makes this method pretty useless (at least directly).

Acceptance-rejection method

- Lets consider

$$f(x) = A \cdot x^3 \cdot e^{-x^2}$$

with $x \in [0,1]$, $A = \frac{2e}{e-2}$



- CDF is given by

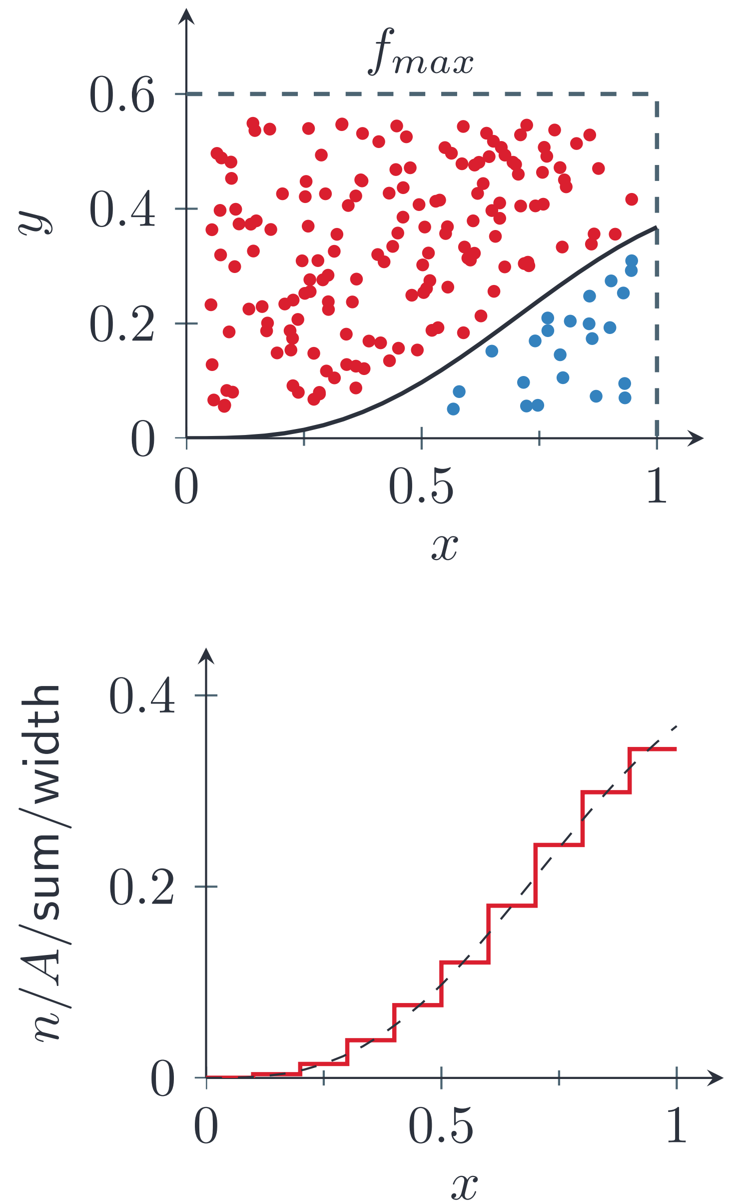
$$F(x) = \frac{N}{2}(x^2 - 1)e^{-x^2}$$

- Since, we do not know F^{-1} we have to find another way to generate x from $f(x)$ distribution
- We will use acceptance-rejection method (do you remember MC integration via hit-or-miss?)

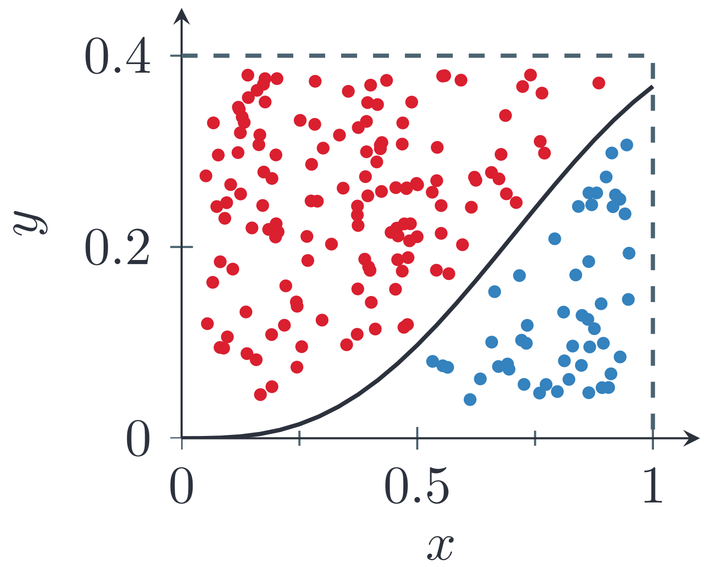
- Evaluate $f_{max} \geq \max(f)$

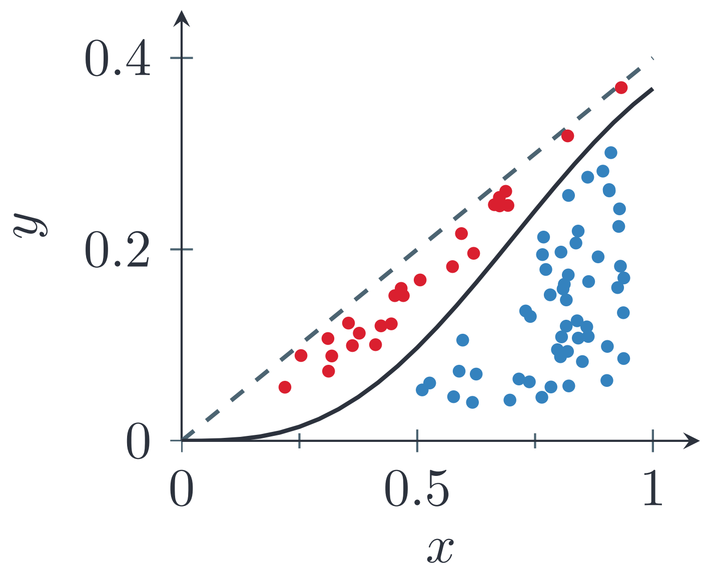
  Note: $f_{max} > \max(f)$ will affect performance, but the result will be still correct

- Generate random x
- Accept x with $P = \frac{f(x)}{f_{max}}$
  - generate a random u from $[0, f_{max}]$
  - accept if $u < f(x)$
- The plot on the right shows the results for $N = 10^5$



- The area under the plot of $f(x)$ is ~ 0.13
- The total area is 0.4
- Thus, only about 30% of points gives contribution to the final distribution
- One can find $g(x)$ for which CDF method is possible and which encapsulates $f(x)$ in given range and generate x according to $g(x)$
- For $g(x) = 0.4x$ the total area is 0.2, so we speed up twice





Acceptance-rejection method - optimization

- Cumulative distribution function for $g(x) = 2x$

$$G(x) = \int_0^x g(t)dt = x^2 \Rightarrow G^{-1}(x) = \sqrt{x}$$

Note: PDF must be normalized to 1 for CDF

- Generate random number $u \in [0, 1]$
- Calculate your $x = G^{-1}(u)$
- Accept x with probability $P = f(x)/g(x)$

instead of using constant f_{max} *we are using* $f_{max}(x) \equiv g(x)$

Quasi-elastic scattering

Building a generator step by step

Quasi-elastic scattering on a free nucleon

Llewellyn-Smith formula

$$\frac{d\sigma}{d|q^2|}\begin{pmatrix}\nu_l + n \to l^- + p \\ \bar{\nu}_l + p \to l^+ + n\end{pmatrix} = \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q^2) \mp B(q^2)\frac{(s-u)}{M^2} + C(q^2)\frac{(s-u)^2}{M^4}\right]$$

Notation

- Constants: M - nucleon mass, G_F - Fermi constant, θ_C - Cabibbo angle,
- $q^2 = (k-k')^2 = (p'-p)^2$ - four-momentum squared, where k, k', p, p' are four-momenta of initial and final lepton, initial and final nucleon
- E_ν - neutrino energy
- $s = (k+k')^2$ and $u = (k-p')^2$ - Mandelstam variables

Quasi-elastic scattering on a free nucleon

Llewellyn-Smith formula

$$\frac{d\sigma}{d|q^2|}\begin{pmatrix}\nu_l + n \to l^- + p \\ \bar{\nu}_l + p \to l^+ + n\end{pmatrix} = \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q^2) \mp B(q^2)\frac{(s-u)}{M^2} + C(q^2)\frac{(s-u)^2}{M^4}\right]$$

General idea

- Having k and p, generate k' and p'
- Calculate q^2 and $(s-u) = 4ME_\nu + q^2 - m^2$ based on generated kinematics
- Calculate cross section
- Repeat N times and the result is given by:

$$\sigma_{total} \sim \frac{1}{N}\sum_{i=1}^{N}\sigma(q_i^2)$$

Generating kinematics

- Lets consider kinematics in center-of-mass system
- Mandelstam s is invariant under Lorentz transformation

$$s = (k+p)^2 = (E+E_p)^2 - (\vec{k}+\vec{p})^2 = (E^* + E_p^*)^2$$

- $\sqrt{s}$ is the total energy in CMS

$$\sqrt{s} = E^* + E_p^* = \sqrt{p^{*2} + m^2} + \sqrt{p^{*2} + M^2}$$

- We will use it to calculate $p*$

Generating kinematics

- Lets do some simple algebra:

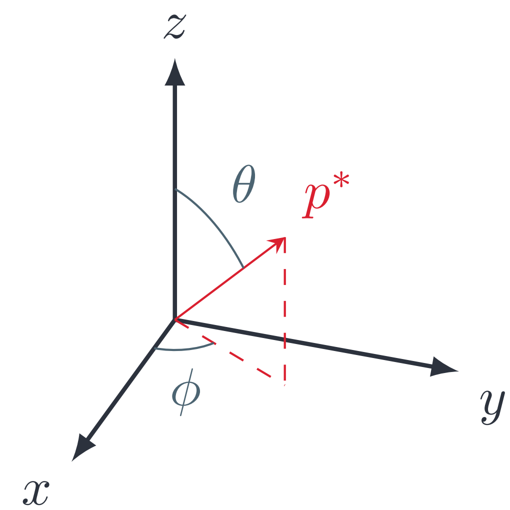
$$\begin{aligned}
\sqrt{s} &= E^* + E_p^* = \sqrt{p^{*2} + m^2} + \sqrt{p^{*^2} + M^2} \\
\sqrt{s} &= E^* + \sqrt{E^{*2} - m^2 + M^2} \\
s &= E^{*2} + E^{*2} - m^2 + M^2 + 2E^* E_p^* \\
s &= 2E^*(E^* + E_p^*) - m^2 + M^2 \\
s &= 2E^*\sqrt{s} - m^2 + M^2 \\
E^* &= \frac{s + m^2 - M^2}{2\sqrt{s}} \\
E_p^* &= \frac{s + M^2 - m^2}{2\sqrt{s}} \text{ (analogously)}
\end{aligned}$$

- After more algebra we get:

$$p^* = \sqrt{E^{*2} - m^2} = \frac{[s - (m - M)^2] \cdot [s - (m + M)^2]}{2\sqrt{s}}$$

Generating kinematics

- We use spherical coordinate system to determine momentum direction in CMS:



$$\vec{p}^{\,*} = p^* \cdot (\sin\theta\cos\phi,\ \ \sin\theta\sin\phi,\ \ \cos\theta)$$

- Generate random angles:

$$\phi = 2\pi \cdot \mathsf{random}[0,1] \Rightarrow \sin\phi,\ \ \cos\phi$$

$$\cos\theta = 2 \cdot \mathsf{random}[0,1] - 1 \Rightarrow \sin\theta,\ \ \cos\theta$$

- All we need to do is to go back to LAB frame

LAB $\leftrightarrows$ CMS

- Lorentz boost in direction $\hat{n} = \frac{\vec{v}}{v}$ of $(t, \vec{r})$:

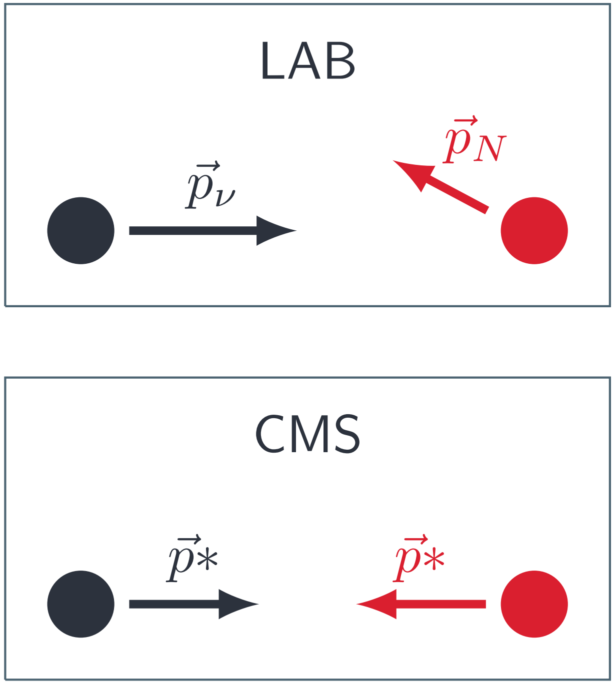
$$
\begin{aligned}
t' &= \gamma\left(t - v\hat{n}\cdot\vec{r}\right) \\
\vec{r}' &= \vec{r} + (\gamma - 1)(\hat{n}\cdot\vec{r})\hat{n} - \gamma t v\hat{n}
\end{aligned}
$$

- In our case

$$
\vec{v} = \frac{\vec{p}_\nu + \vec{p}_N}{E_\nu + E_N}
$$

- Boost from LAB to CMS in $\vec{v}$ direction

- Boost from CMS to LAB in $-\vec{v}$ direction



Calculating cross section

Llewellyn-Smith formula

$$\frac{d\sigma}{d|q^2|}\begin{pmatrix}\nu_l + n \to l^- + p \\ \bar{\nu}_l + p \to l^+ + n\end{pmatrix} = \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q^2) \mp B(q^2)\frac{(s-u)}{M^2} + C(q^2)\frac{(s-u)^2}{M^4}\right]$$

Calculation

- Once we have p' and k' in LAB frame we can calculate q^2 and $(s-u)$
- Once we have q^2 we can calculate $A(q^2)$, $B(q^2)$, $C(q^2)$
- We have everything to calculate cross section
- Do we? Or maybe we are still missing something?

Calculating cross section

Llewellyn-Smith formula

$$\frac{d\sigma}{d|q^2|}\begin{pmatrix}\nu_l + n \to l^- + p \\ \bar{\nu}_l + p \to l^+ + n\end{pmatrix} = \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q^2) \mp B(q^2)\frac{(s-u)}{M^2} + C(q^2)\frac{(s-u)^2}{M^4}\right]$$

Calculation

- Once we have p' and k' in LAB frame we can calculate q^2 and $(s-u)$
- Once we have q^2 we can calculate $A(q^2)$, $B(q^2)$, $C(q^2)$
- We have everything to calculate cross section
- Do we? Or maybe we are still missing something?

We change the variable we integrate over! We need Jacobian!

Calculating cross section

- Express q^2 in terms of angle:

$$q^2 = (k - k')^2 = m^2 - 2kk' = m^2 - 2EE' + 2|\vec{k}||\vec{k}'|\cos\theta$$

- Thus, the Jacobian is given by:

$$dq^2 = 2|\vec{k}||\vec{k}'|d(\cos\theta)$$

Note: must be calculated in CMS

- Total cross section is given by:

$$\sigma = \int_{-1}^{1} \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q^2) \mp B(q^2)\frac{(s-u)}{M^2} + C(q^2)\frac{(s-u)^2}{M^4}\right] 2|\vec{k}||\vec{k}'|d\cos\theta$$

$$\sigma_{MC} = \frac{2}{N}\sum_{i=1}^{N} \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q_i^2) \mp B(q_i^2)\frac{(s_i-u_i)}{M^2} + C(q_i^2)\frac{(s_i-u_i)^2}{M^4}\right] 2|\vec{k}_i||\vec{k}'_i|$$

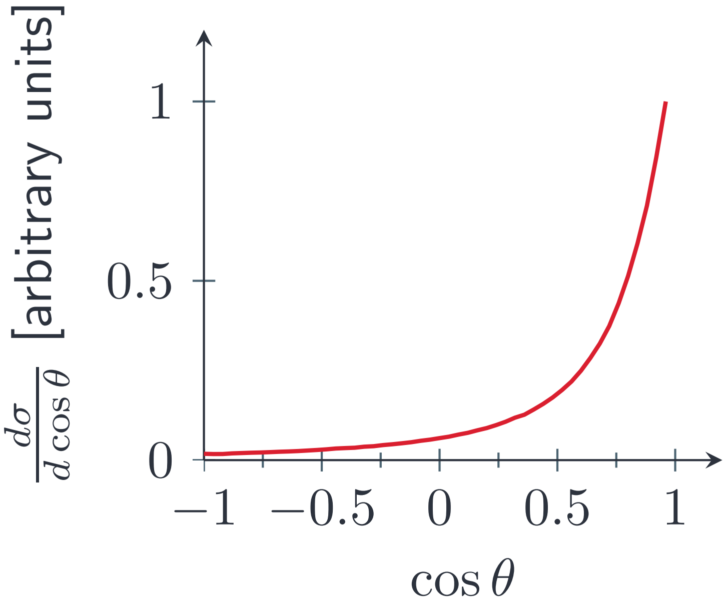
- We want to avoid any sharp peaks
- They affect our efficiency and accuracy
- Lets change variable once again:

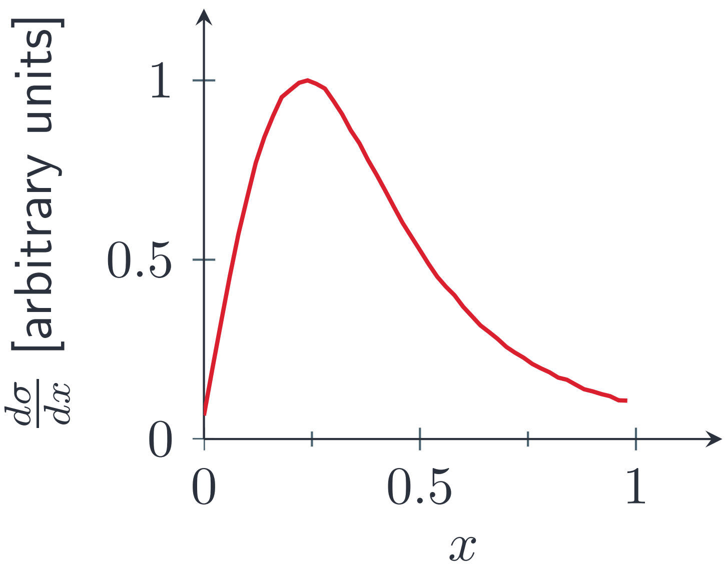
$$\cos\theta = 1 - 2x^2$$

where $x \in [0, 1]$

- Note extra Jacobian and new integration limits

$$2\int_{-1}^{1} d(\cos\theta) \rightarrow \int_{1}^{0} dx(-4x) \rightarrow \int_{0}^{1} 4x dx$$





Calculating cross section

- Finally, the cross section is given by:

$$\sigma = \int_0^1 \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q^2) \mp B(q^2)\frac{(s-u)}{M^2} + C(q^2)\frac{(s-u)^2}{M^4}\right] 2|\vec{k}||\vec{k}'|4x dx$$

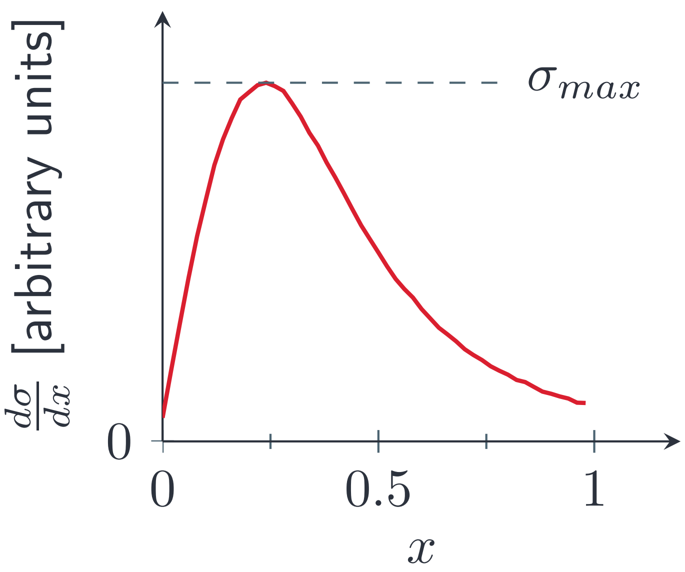
$$\sigma_{MC} = \frac{1}{N}\sum_{i=1}^{N} \frac{M^2 G_F^2 \cos\theta_C}{8\pi E_\nu^2}\left[A(q_i^2) \mp B(q_i^2)\frac{(s_i-u_i)}{M^2} + C(q_i^2)\frac{(s_i-u_i)^2}{M^4}\right] 2|\vec{k}_i||\vec{k}_i'|4x$$

- In conclusion: do some kinematics and some boosts between CMS and LAB, change integration variable several times... and you are ready to calculate total cross section
- Now we need to generate some events. We want them to be distributed according to our cross section formula.

Generating events

- Generate $x \in [0:1]$
- Do kinematics

$$\begin{array}{rcl} x & \rightarrow & \cos\theta \\ \cos\theta & \rightarrow & k'^*, p'^* \\ k'^*, p'^* & \rightarrow & k', p' \\ & \vdots & \end{array}$$



- Calculate cross section σ
- Accept an event with the probability given by

$$P = \frac{\sigma}{\sigma_{max}}$$

- And you almost have you MC neutrino-event generator, just a few more steps...

A few more steps

- add other dynamics: resonance pion production, deep inelastic scattering...
- add support for nucleus as a target
- if you have nucleus add some two-body current interactions
- if you have nucleus add some nuclear effects: Pauli blocking, final state interactions, formation zone...
- add support for neutrino beam
- add support for detector geometry
- add some interface to set up simulations parameters and saving the output
- and your MC is done!

Tutorial: Monte Carlo methods

PRNG

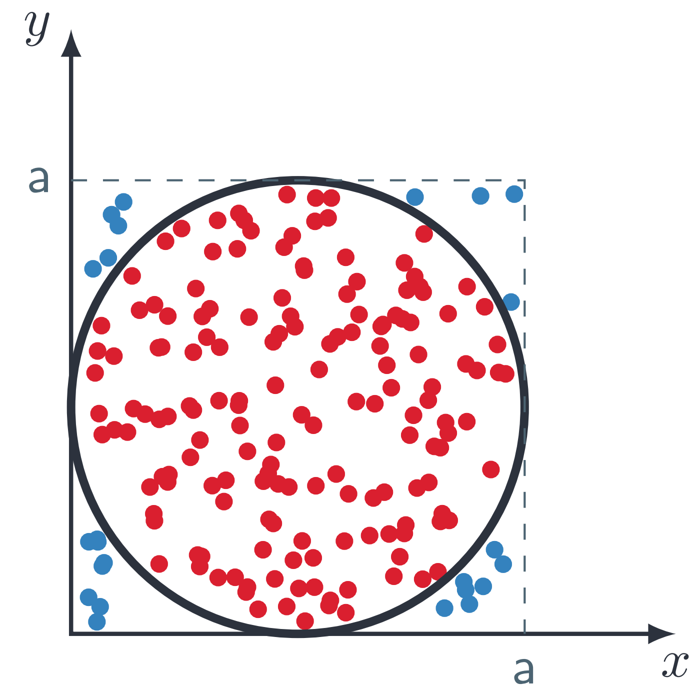
- You can use whatever random number generator you want
- If you are using C++ you may consider using PRNG class, which wraps up mersenne twister engine [link to PRNG.h]
- Usage:

```
const PRNG random (min, max);
random.generate00(); // returns RN from (min, max)
random.generate01(); // returns RN from (min, max]
random.generate10(); // returns RN from [min, max)
random.generate11(); // returns RN from [min, max]
```

Task 1: evaluate π

Evaluate π using MC method

- get N random points from a square
- count how many points are inside a circle
- calculate π
- check how the results depends on N



Lets consider the following function:

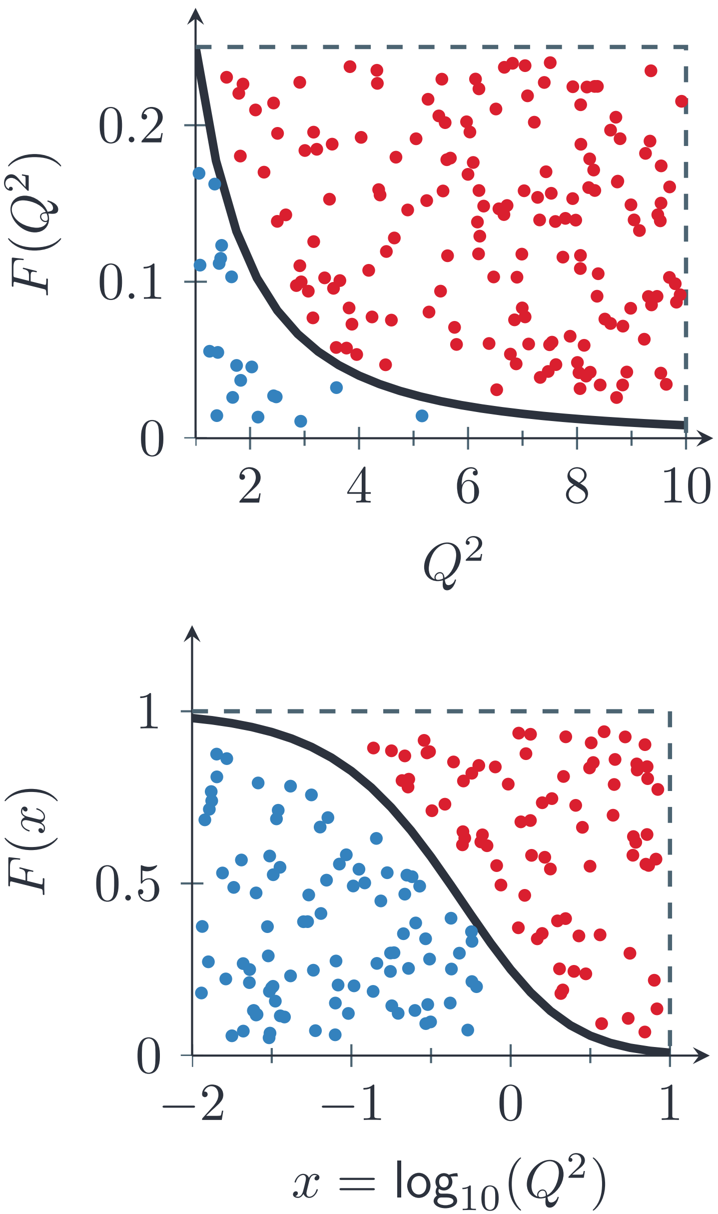
$$F(Q^2) = \frac{1}{(1+Q^2)^2}$$

a) Integrate this function over Q^2 using hit-or-miss method

b) Integrate this function over $x = \log_{10}(Q^2)$ using the same method

c) Compare efficiency

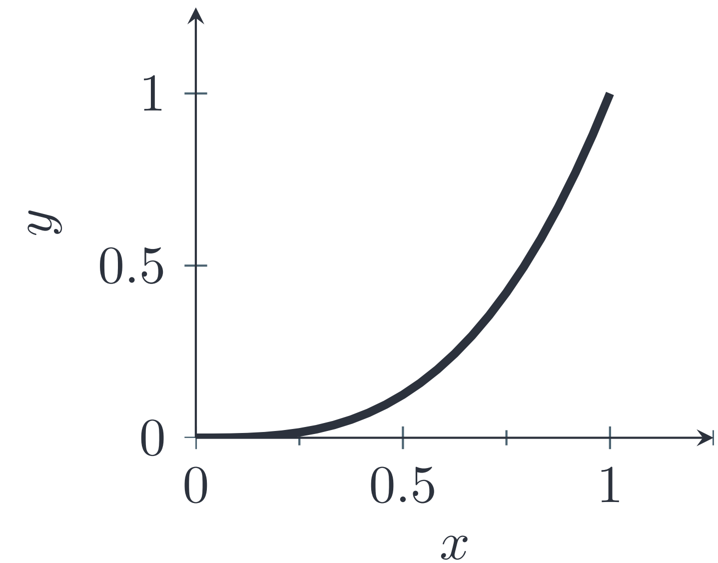
d) Integrate this function using crude method



Task 3: generating number from distribution

Write a program to generate random numbers from $[0, 1]$ according to the following distribution:

$$f(x) = x^3$$



a) using cumulative distribution function

b) using acceptance-rejection method (consider substitution to get better performance)

Task 4*: neutrino-electron scattering

- For $E_\nu >> m_e$, the cross section for $\nu_\mu - e$ scattering can be approximated by:

$$\frac{d\sigma}{dy} = \frac{G_F^2 s}{\pi}\left[A^2 + B^2 \cdot (1-y)^2\right]$$

where G_F - Fermi weak coupling constant, s - Mandelstam variable, $y \equiv \frac{T_e}{E_\nu}$ with T_e - electron kinetic energy, $A = \frac{1}{2} - \sin^2\theta_W$, $B = \sin^2\theta_W$, θ_W - Weinberg angle

a) Write a program to calculate total cross section for given neutrino energy

b) Using results from a), generate $\frac{d\sigma}{dT_e}$ distribution

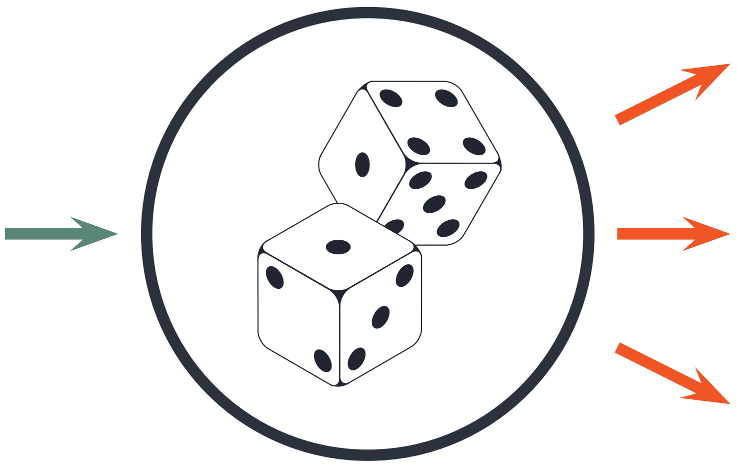
c) Using results from a), generate $\frac{d\sigma}{d\cos\theta}$ distribution, hint:

$$T_e = \frac{2m_e E_\nu^2 \cos^2\theta}{(m_e + E_\nu)^2 - E_\nu^2\cos^2\theta} \approx 2m_e \frac{\cos^2\theta}{1-\cos^2\theta}$$

Monte Carlo neutrino event generators

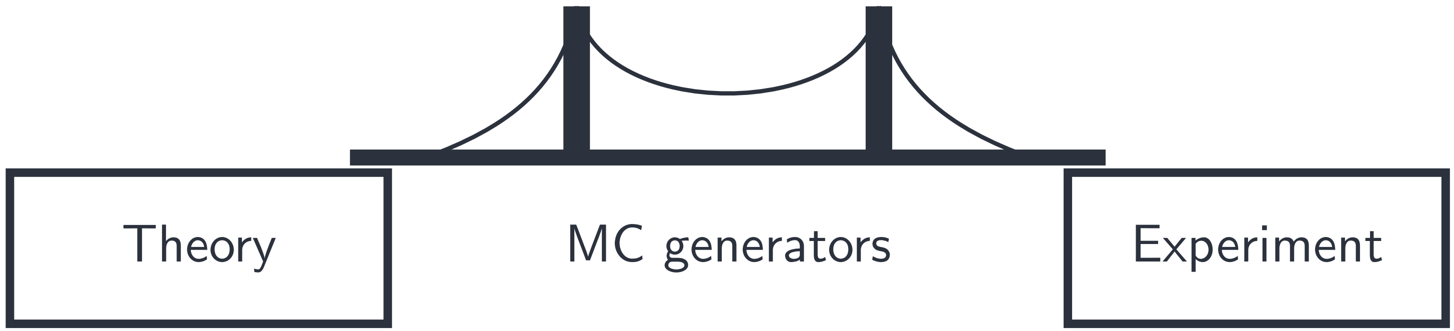
Monte Carlo event generators

- Monte Carlo generators simulate interactions
- Physicists have been using them since ENIAC
- Some common generators used in neutrino community:



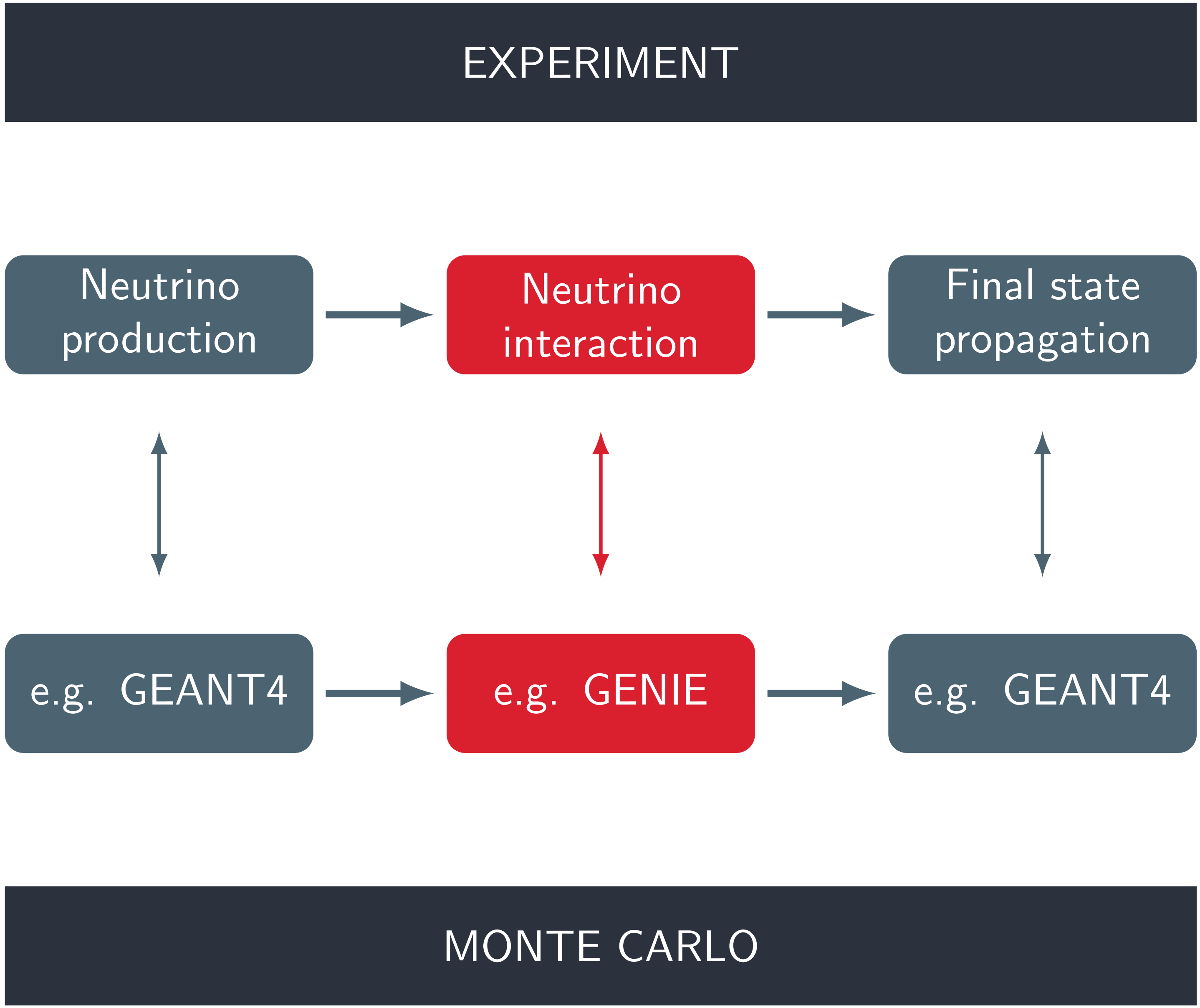
  - transport of particles through matter: **Geant4**, **FLUKA**
  - high-energy collisions of elementary particles: **PYTHIA**
  - neutrino interactions: **GENIE**, **GIBUU**, **NEUT**, **NUANCE**, **NuWro**

Why do we need them?



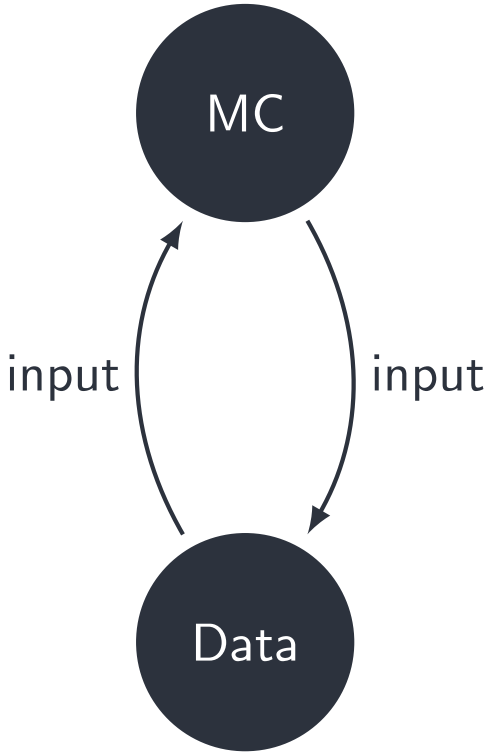
- Monte Carlo event generators connect experiment (what we see) and theory (what we think we should see)
- Any neutrino analysis relies on MC generators
- From neutrino beam simulations, through neutrino interactions, to detector simulations
- Used to evaluate systematic uncertainties, backgrounds, acceptances...

Why do we need them?



"You use Monte Carlo until you understand the problem"
Mark Kac

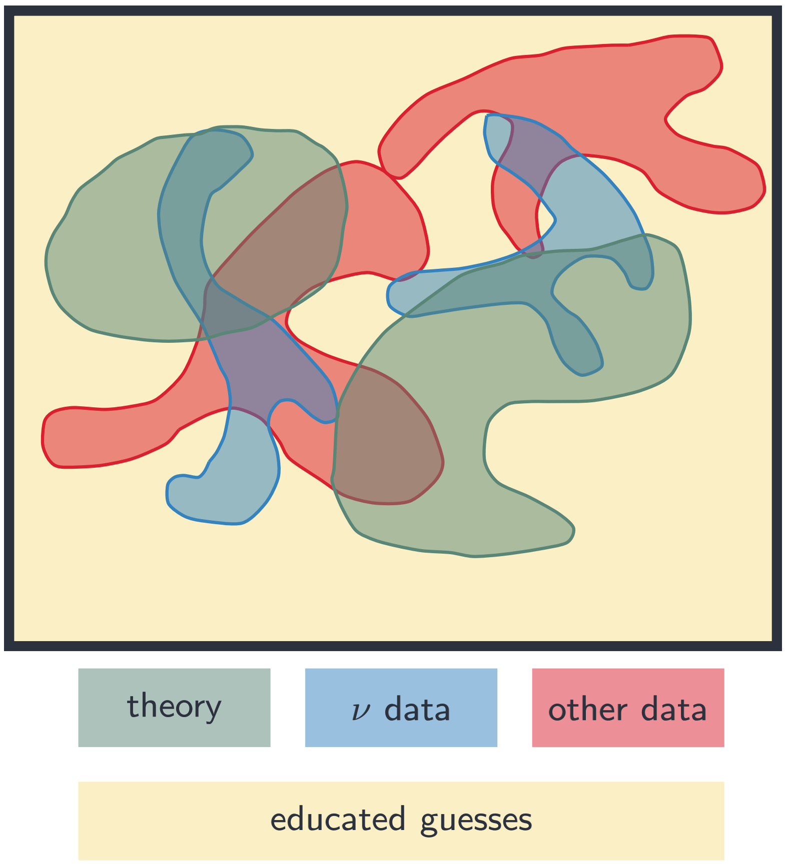
- In perfect world MC generators would contain "pure" theoretical models
- In real world theory does not cover everything
- Neutrino and non-neutrino data are used to tune generators



How to build generator

INGREDIENTS:

Phase space

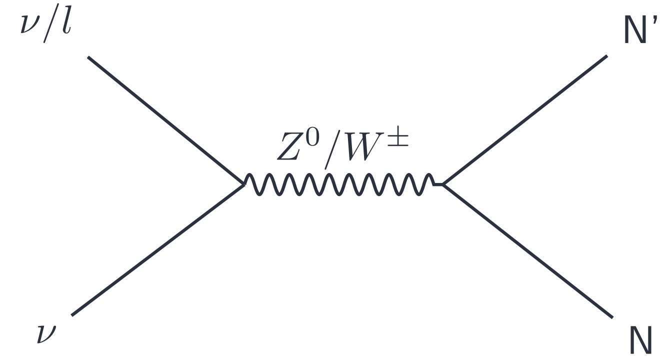


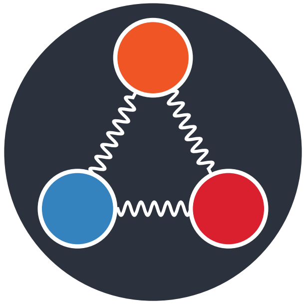
RECIPE:

Neutrino interactions: free nucleon

(Quasi-)elastic scattering

- Llewellyn-Smith model is usually used for charged current quasi-elastic scattering
- Not much difference here between generators (but default parameters)





- Nucleon structure is parametrized by form factors

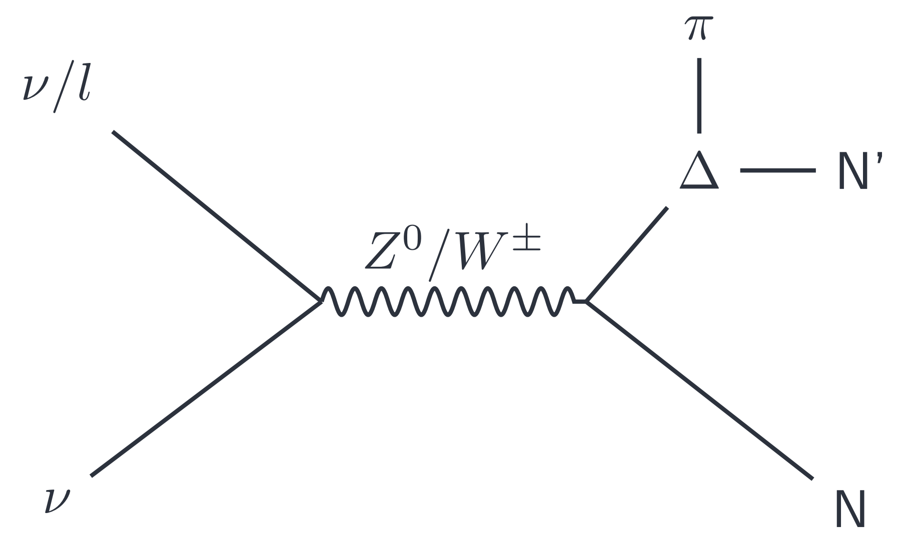
- Vector $\rightarrow$ Conserved Vector Current (CVC)
- Pseudo-scalar $\rightarrow$ Partially Conserved Axial Current (PCAC)
- Axial $\rightarrow$ dipole form with one free parameter (axial mass, M_A)

Rein-Sehgal model

TABLE I

Nucleon Resonances below 2 GeV/c² according to Ref. [4]

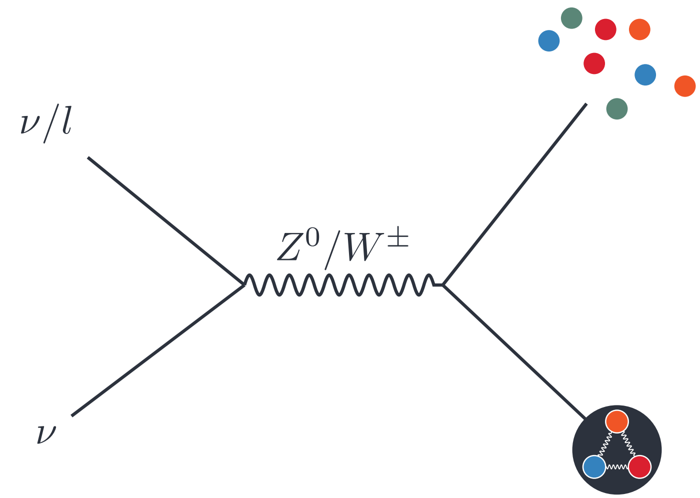
Resonance Symbol[a]	Central mass value M [MeV/c²]	Total with Γ_0[MeV]	Elasticity $x_E = \pi\mathcal{N}$ branching ratio	Quark-Model/ SU_6-assignment
$P_{33}(1234)$	1234	124	1	$^4(10)_{3/2}\,[56, 0^+]_0$
$P_{11}(1450)$	1450	370	0.65	$^2(8)_{1/2}\,[56, 0^+]_2$
$D_{13}(1525)$	1525	125	0.56	$^2(8)_{3/2}\,[70, 1^-]_1$
$S_{11}(1540)$	1540	270	0.45	$^2(8)_{1/2}\,[70, 1^-]_1$
$S_{31}(1620)$	1620	140	0.25	$^2(10)_{1/2}\,[70, 1^-]_1$
$S_{11}(1640)$	1640	140	0.60	$^4(8)_{1/2}\,[70, 1^-]_1$
$P_{33}(1640)$	1640	370	0.20	$^4(10)_{3/2}\,[56, 0^+]_2$
$D_{13}(1670)$	1670	80	0.10	$^4(8)_{3/2}\,[70, 1^-]_1$
$D_{15}(1680)$	1680	180	0.35	$^4(8)_{5/2}\,[70, 1^-]_1$
$F_{15}(1680)$	1680	120	0.62	$^2(8)_{5/2}\,[56, 2^+]_2$
$P_{11}(1710)$	1710	100	0.19	$^2(8)_{1/2}\,[70, 0^+]_2$
$D_{33}(1730)$	1730	300	0.12	$^2(10)_{3/2}\,[70, 1^-]_1$
$P_{13}(1740)$	1740	210	0.19	$^2(8)_{3/2}\,[56, 2^+]_2$
$P_{31}(1920)$	1920	300	0.19	$^4(10)_{1/2}\,[56, 2^+]_2$
$F_{35}(1920)$	1920	340	0.15	$^4(10)_{5/2}\,[56, 2^+]_2$
$F_{37}(1950)$	1950	340	0.40	$^4(10)_{7/2}\,[56, 2^+]_2$
$P_{33}(1960)$	1960	300	0.17	$^4(10)_{3/2}\,[56, 2^+]_2$
$F_{17}(1970)$	1970	325	0.06	$^4(8)_{7/2}\,[70, 2^+]_2$



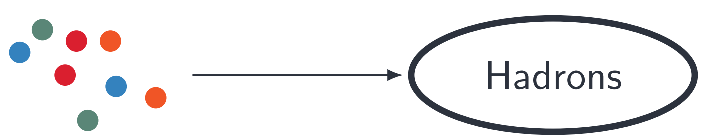
- Rein-Sehgal model describes single pion production through baryon resonances below $W = 2$ GeV
- It is used by GENIE and NEUT
- However, GENIE includes only 16 resonances and interference between them is neglected

Deep inelastic scattering [DIS]

- Quark-parton model is used for deep inelastic scattering
- Bodek-Young modification to the parton distributions at low Q^2 is included by most generators



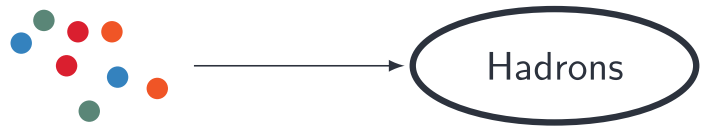
Hadronization



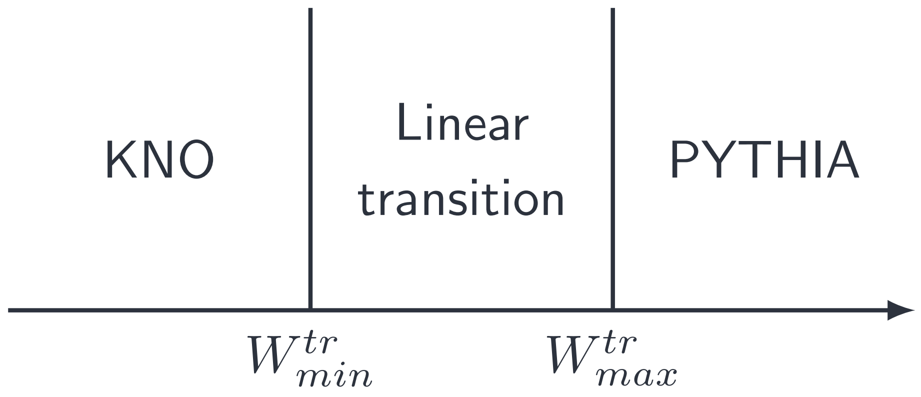
- Hadronization is the process of formation hadrons from quarks
- Pythia is widely used at high invariant masses

Andreopoulos-Gallagher-Kehayias-Yang model

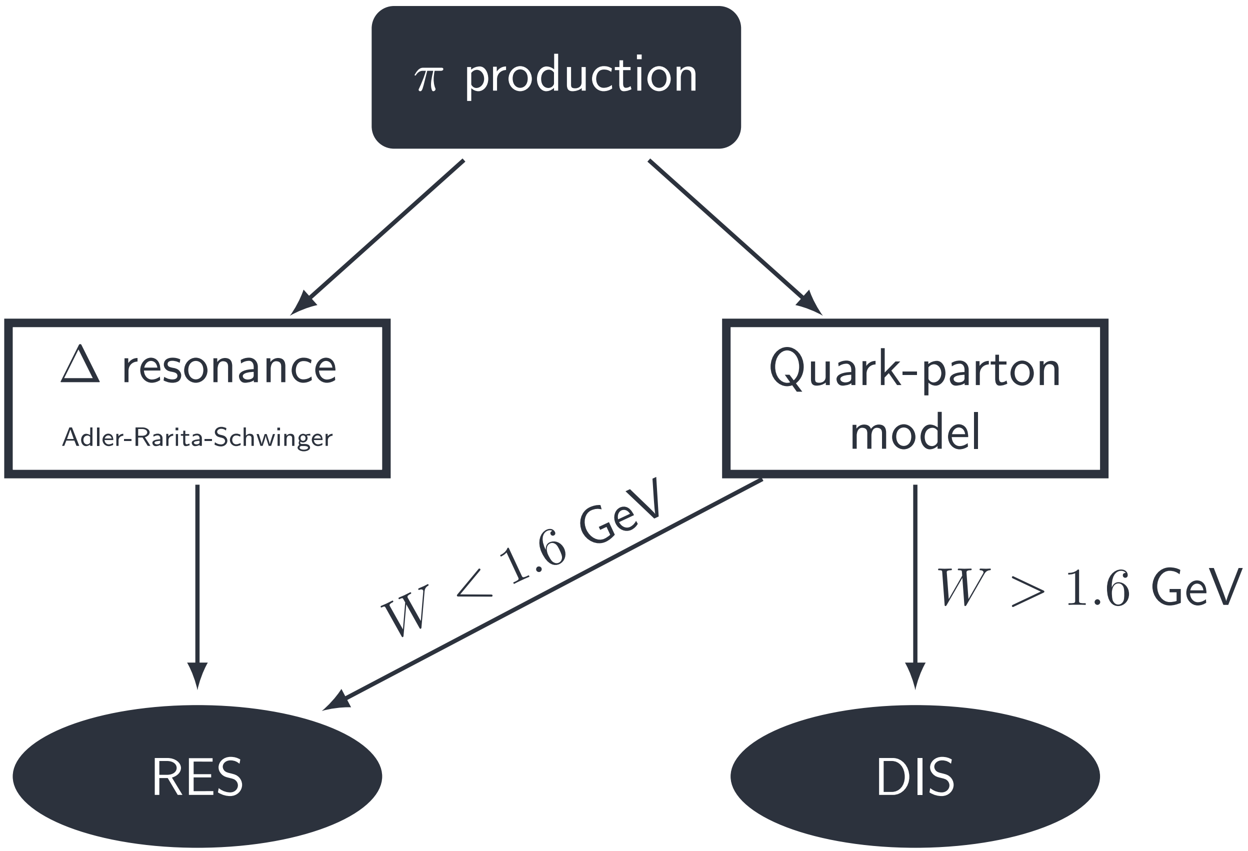
- AGKY hadronization model is used in GENIE



- It includes phenomenological description of the low invariant mass based on Koba-Nielsen-Olesen (KNO) scaling
- Pythia is used for the high invariant mass
- The smooth transition between two models is made in a window $W \in [2.3, 3.0]$ GeV



Pion production in NuWro



RES/DIS distinguish is arbitrary for each MC generator!

Transition region

- We factorized the reality to RES and DIS
- We must be careful to avoid double counting
- The smooth transition between RES and DIS is performed by each generator (but in slightly different way)
- E.g. in GENIE:

$$\frac{d^2\sigma^{RES}}{dQ^2dW} = \sum_k \left(\frac{d^2\sigma^{R-S}}{dQ^2dW}\right)_k \cdot \Theta(W_{cut} - W)$$

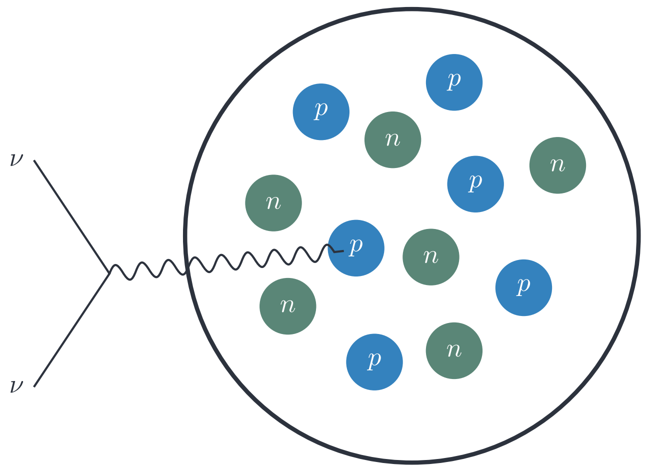
$$\frac{d^2\sigma^{DIS}}{dQ^2dW} = \frac{d^2\sigma^{DIS,BY}}{dQ^2dW} \cdot \Theta(W - W_{cut}) + \frac{d^2\sigma^{DIS,BY}}{dQ^2dW} \cdot \Theta(W_{cut} - W) \cdot \sum_m f_m$$

where k - sum over resonances in Rein-Sehgal model, m - sum over multiplicity, $f_m = R_m \cdot P_m$ with P_m - probability of given multiplicity (taken form hadronization model), R_m - tunable parameter

Neutrino interactions: nucleus

Impulse approximation

- In impulse approximation neutrino interacts with a single nucleon



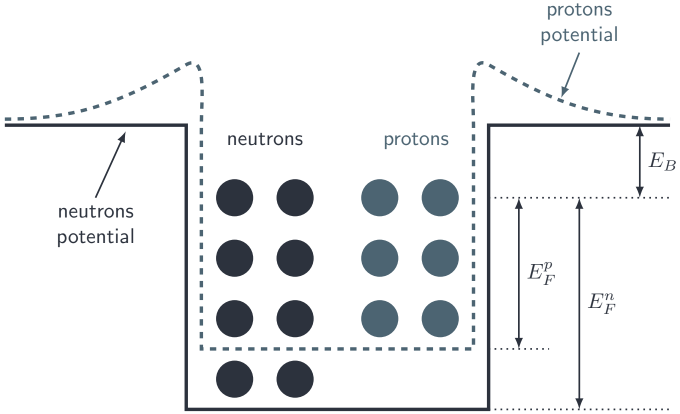
- If $|\vec{q}|$ is low the impact area usually includes many nucleons
- For high $|\vec{q}|$ IA is justified
- Squares of transition matrices are summed up and interference terms are neglected

$$\sigma^A = \sum_{i=1}^{Z} \sigma_p + \sum_{i=1}^{A-Z} \sigma_n$$

- High $|\vec{q}|$ means more than 400 MeV. However, IA is always assumed

Fermi gas

Nucleons move freely within the nuclear volume in constant binding potential.

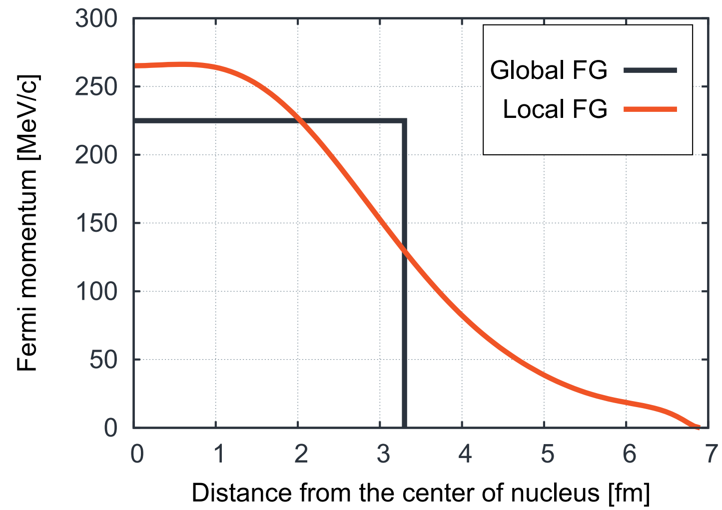


Global Fermi Gas

$$p_F = \frac{\hbar}{r_0}\left(\frac{9\pi N}{4A}\right)^{1/3}$$

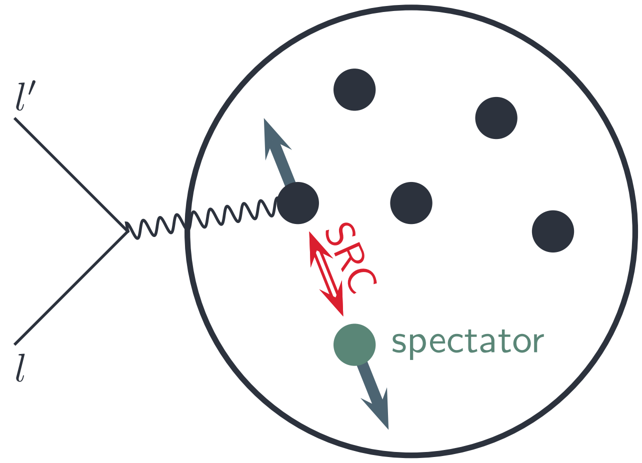
Local Fermi Gas

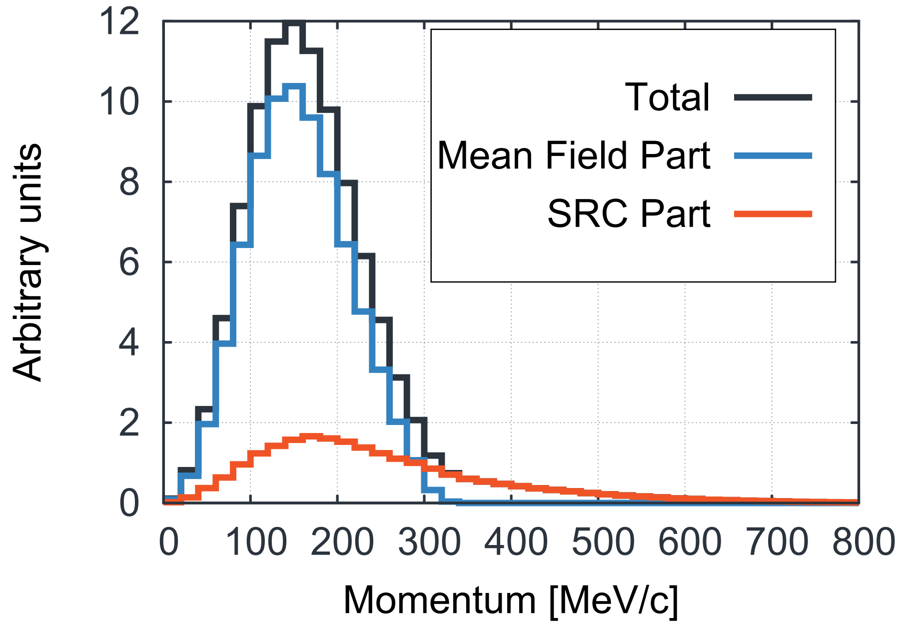
$$p_F(r) = \hbar\left(3\pi^2\rho(r)\frac{N}{A}\right)^{1/3}$$

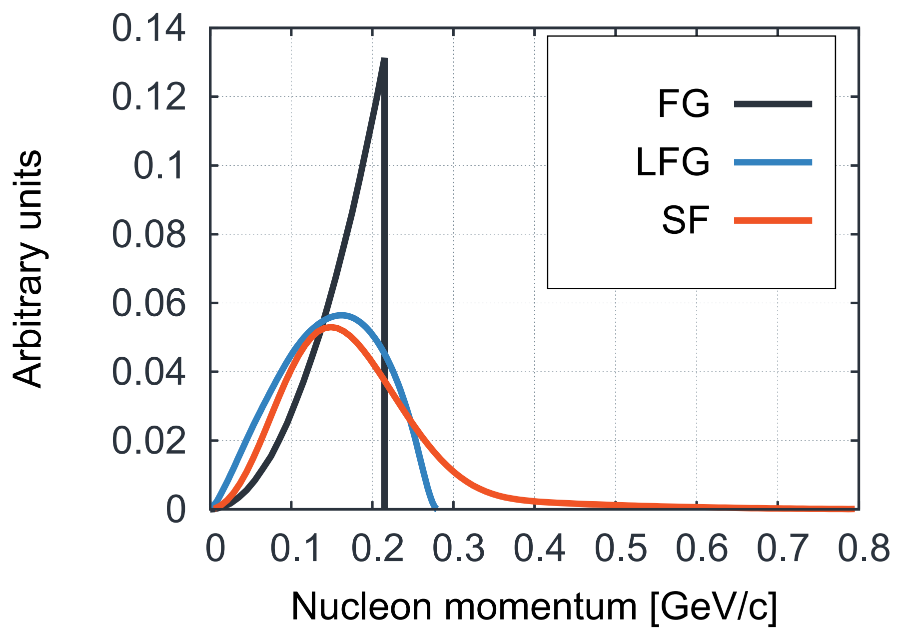


The probability of removing of a nucleon with momentum $\vec{p}$ and leaving residual nucleus with excitation energy E.

$$P(\vec{p}, E) = P_{MF}(\vec{p}, E) + P_{corr}(\vec{p}, E)$$



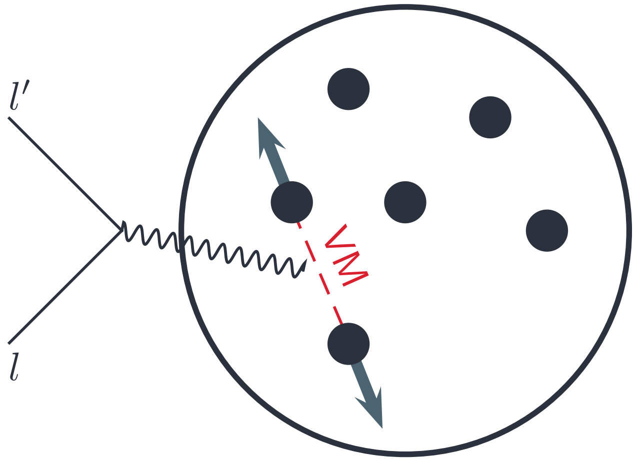




Two Body Current

2 particles - 2 holes (2p-2h)

Meson Exchange Current (MEC)



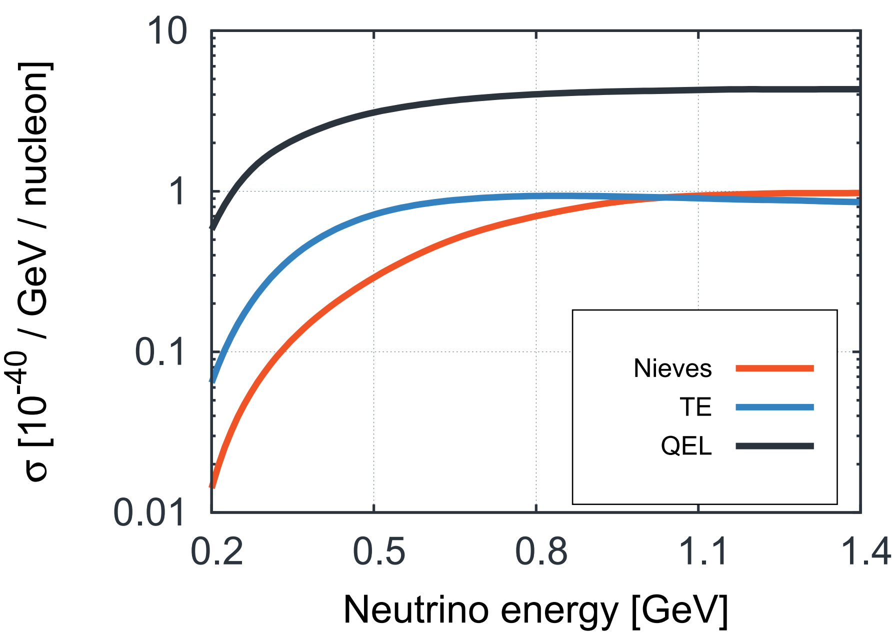
Models in generators

- Nieves model (GENIE - coming soon, NEUT, NuWro)
- Transverse Enhancement (TE) model by Bodek (NuWro)
- Dytman model (GENIE)

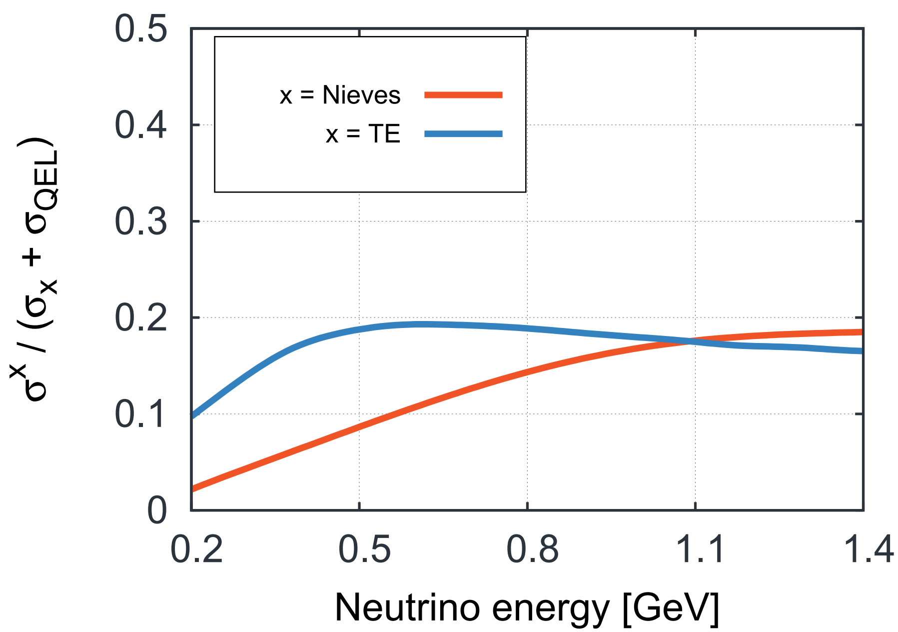
Two-body current interactions

- Nieves model is microscopic calculation
- TE model introduce $2p - 2h$ contribution by modification of the vector magnetic form factors

Total MEC cross section

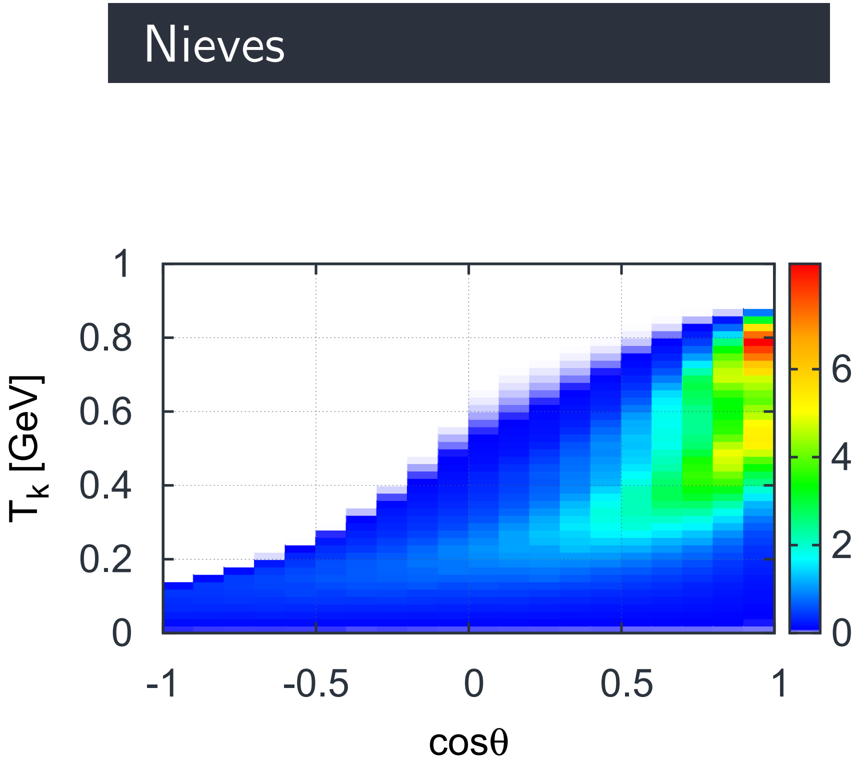


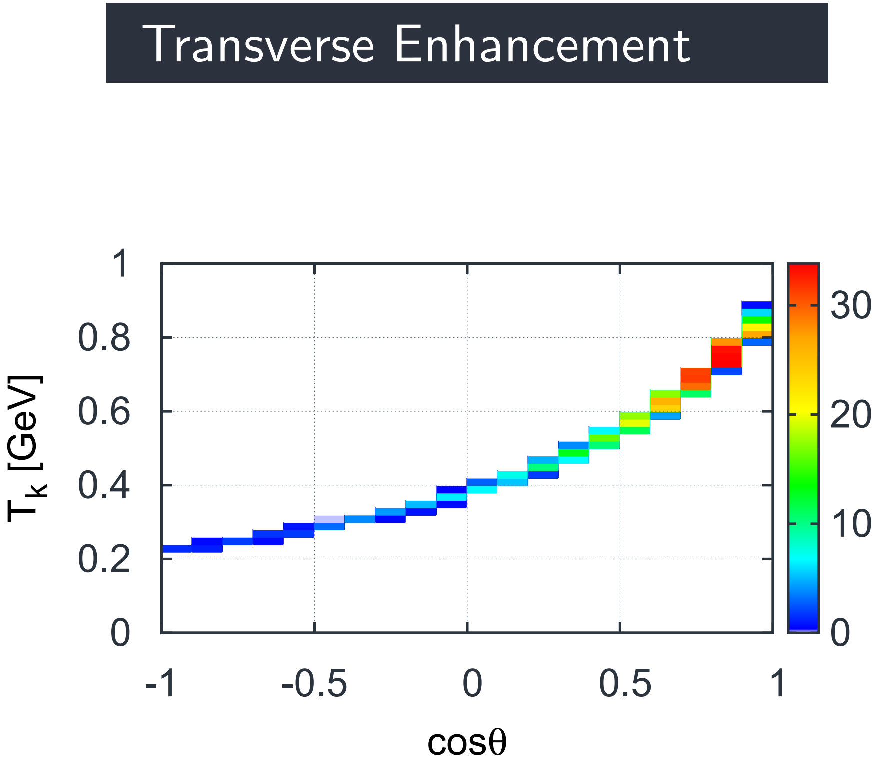
MEC / (QEL + MEC)



Two-body current interactions

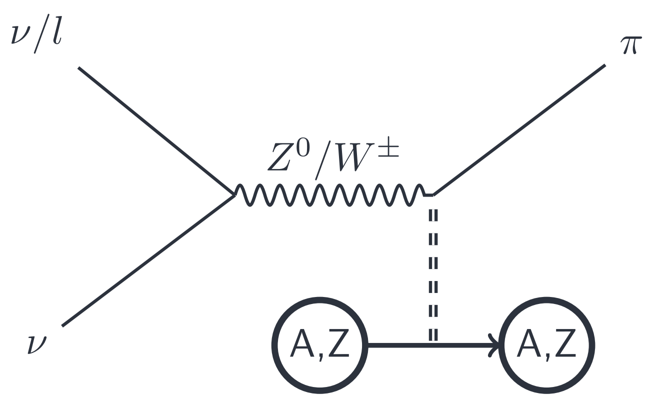
- Both models provide only the inclusive double differential cross section for the final state lepton
- Final nucleons momenta are set isotropically in CMS





Coherent pion production

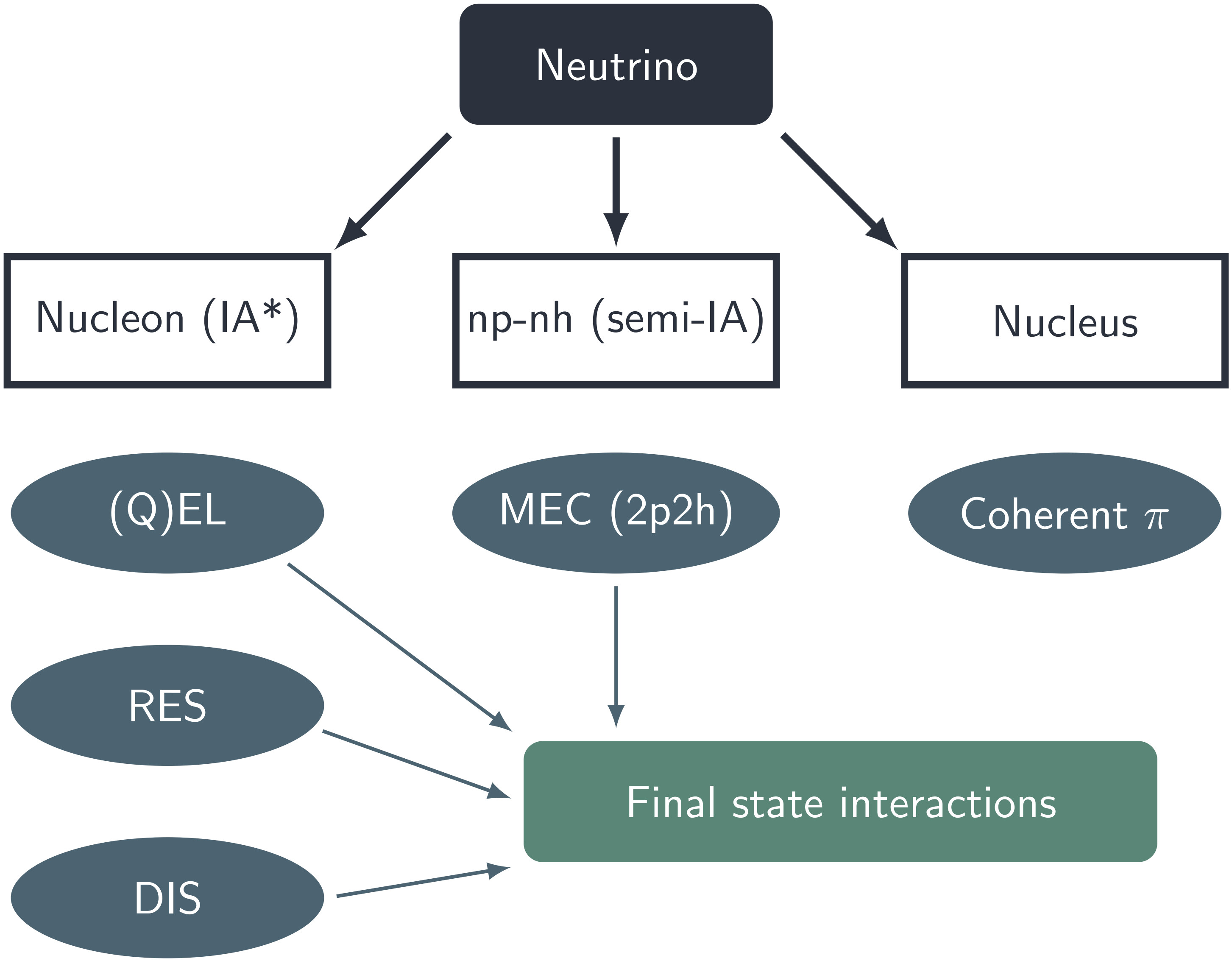
- Rein-Sehgal model is commonly used for coherent pion production
- Note: it is different model than for RES
- Berger-Sehgal model replaces RS (NuWro, GENIE - coming soon)



Comments

- In COH the residual nucleus is left in the same state (not excited)
- The interaction occurs on a whole nucleus - no final state interactions

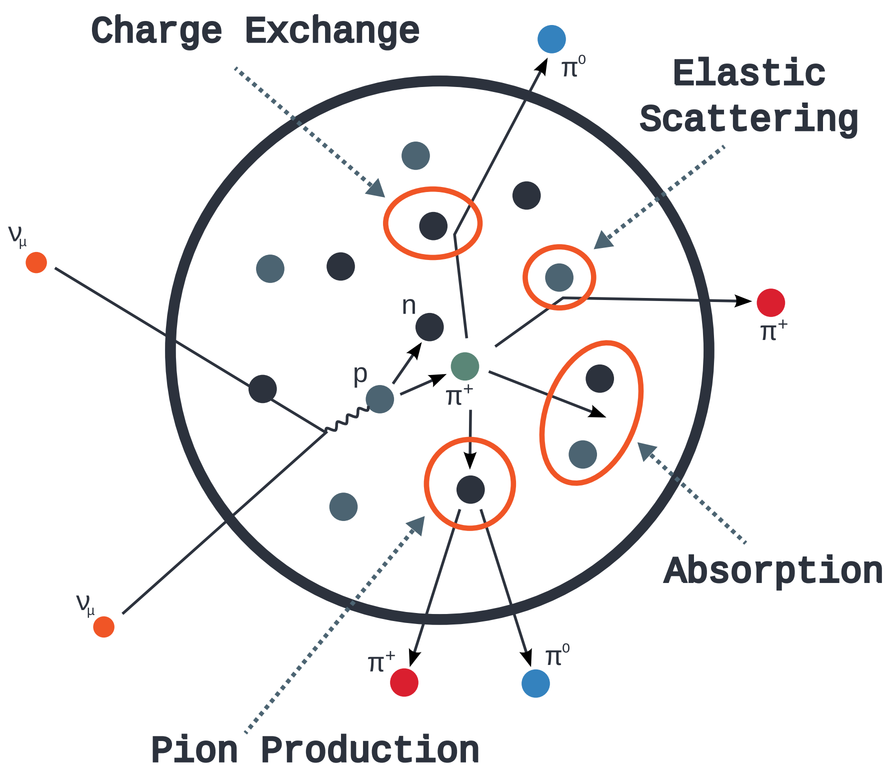
Neutrino interactions - summary



*IA = Impulse Approximation

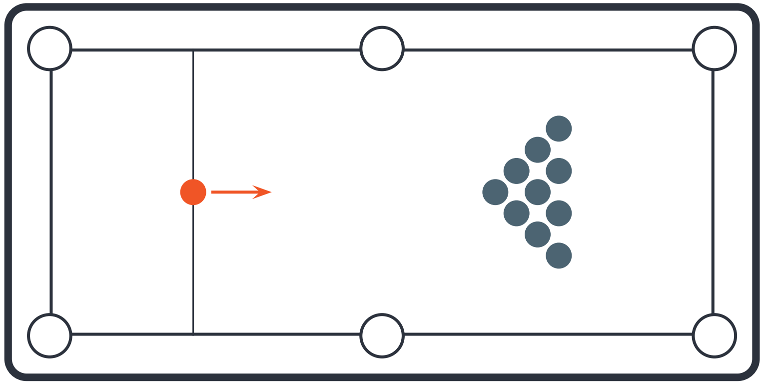
Final state interactions

FSI describe the propagation of particles created in a primary neutrino interaction through nucleus



All MC generators (but GIBUU) use intranuclear cascade model

- In INC model particles are assumed to be classical and move along the straight line.



- The probability of passing a distance λ (small enough to assume constant nuclear density) without any interaction is given by:

$$P(\lambda) = e^{-\lambda/\tilde{\lambda}}$$

$\tilde{\lambda} = (\sigma\rho)^{-1}$ - mean free path

σ - cross section

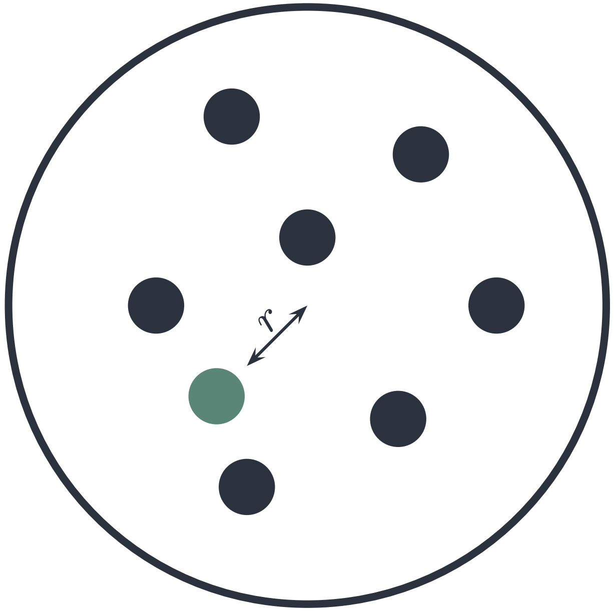
ρ - nuclear density

Can be easily handled with MC methods.

The algorithm for intranuclear cascade

Calculate:

$$\tilde{\lambda}(r) = [\sigma\rho(r)]^{-1}$$



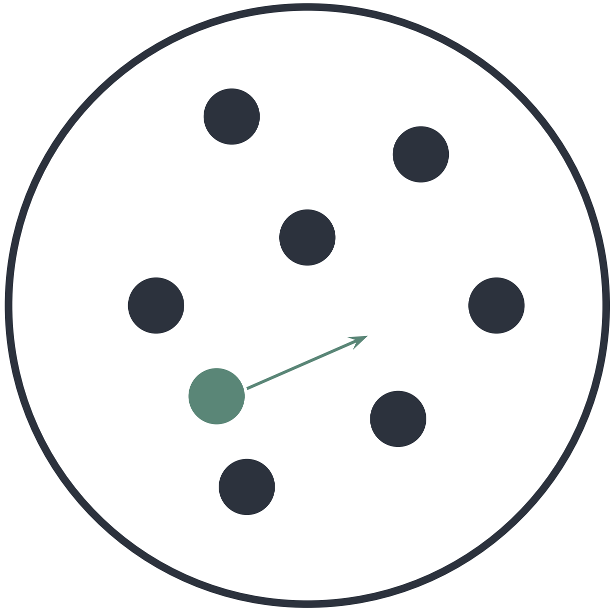
The algorithm for intranuclear cascade

Calculate:

$\tilde{\lambda}(r) = [\sigma\rho(r)]^{-1}$

$\lambda = \tilde{\lambda} \cdot \ln(P)$

$P = \text{rand}[0, 1]$



The algorithm for intranuclear cascade

Calculate:

$\tilde{\lambda}(r) = [\sigma\rho(r)]^{-1}$

→

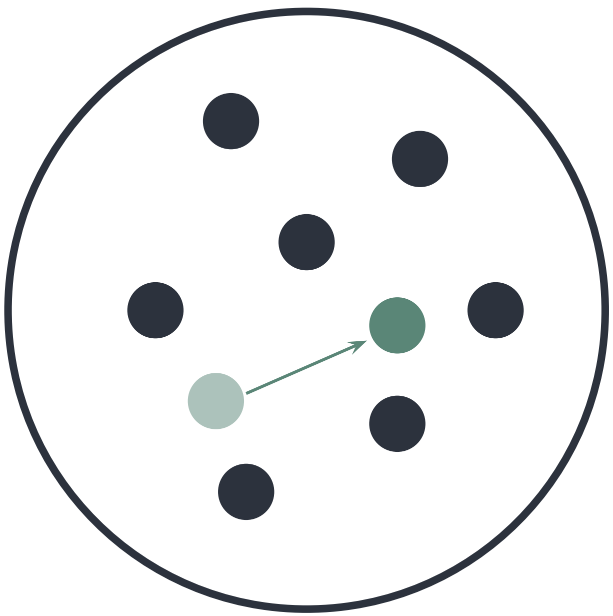
$\lambda = \tilde{\lambda} \cdot \ln(P)$

$P =$ rand$[0,1]$

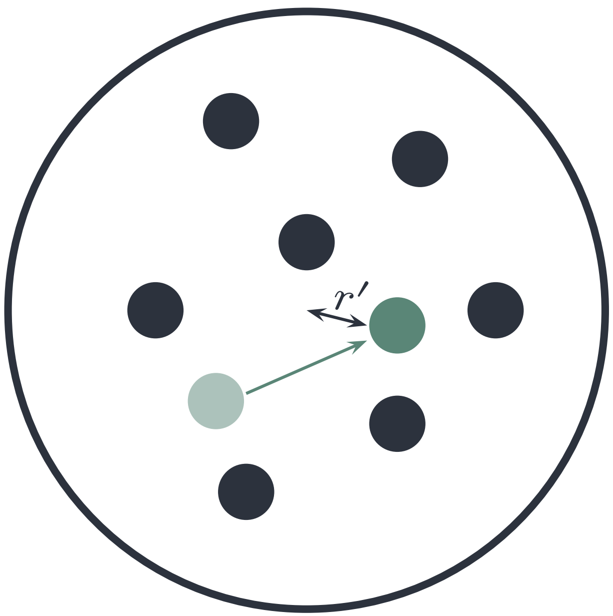
↓

move particle by

min(λ, λ_{max})



The algorithm for intranuclear cascade



Calculate:

$\tilde{\lambda}(r) = [\sigma\rho(r)]^{-1}$

→

$\lambda = \tilde{\lambda} \cdot \ln(P)$

$P =$ rand$[0, 1]$

↓

move particle by

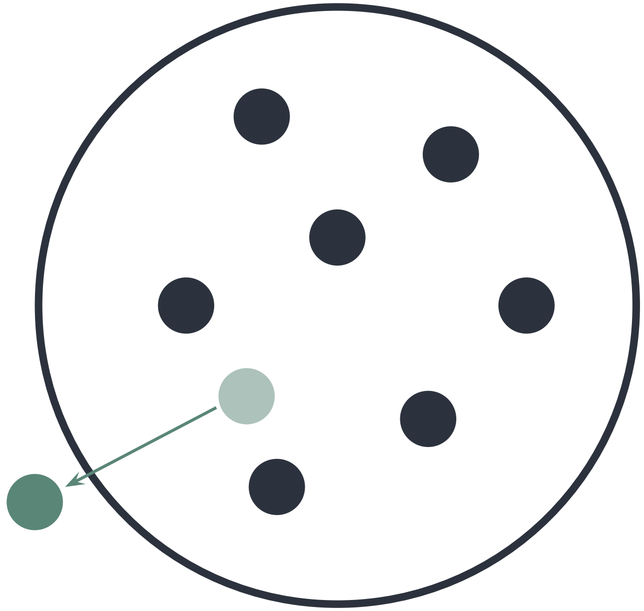
min(λ, λ_{max})

←

Check $r' > R$

R - nucleus radius

The algorithm for intranuclear cascade



Calculate:
$\tilde{\lambda}(r) = [\sigma\rho(r)]^{-1}$

→

$\lambda = \tilde{\lambda} \cdot \ln(P)$
$P =$ rand$[0, 1]$

↓

move particle by
min(λ, λ_{max})

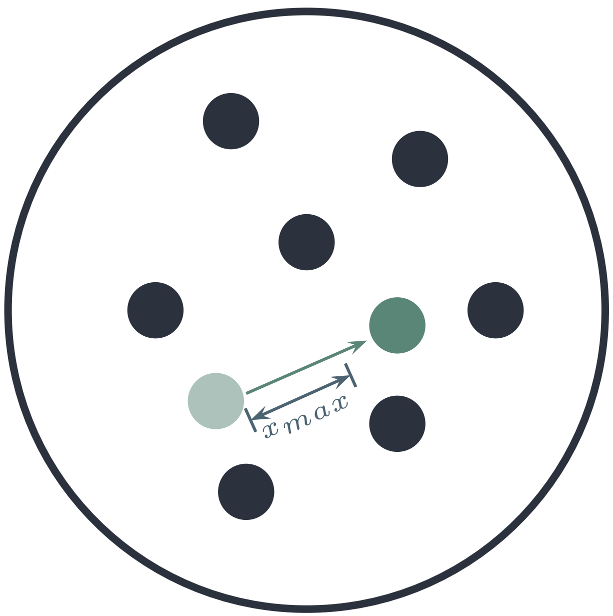
←

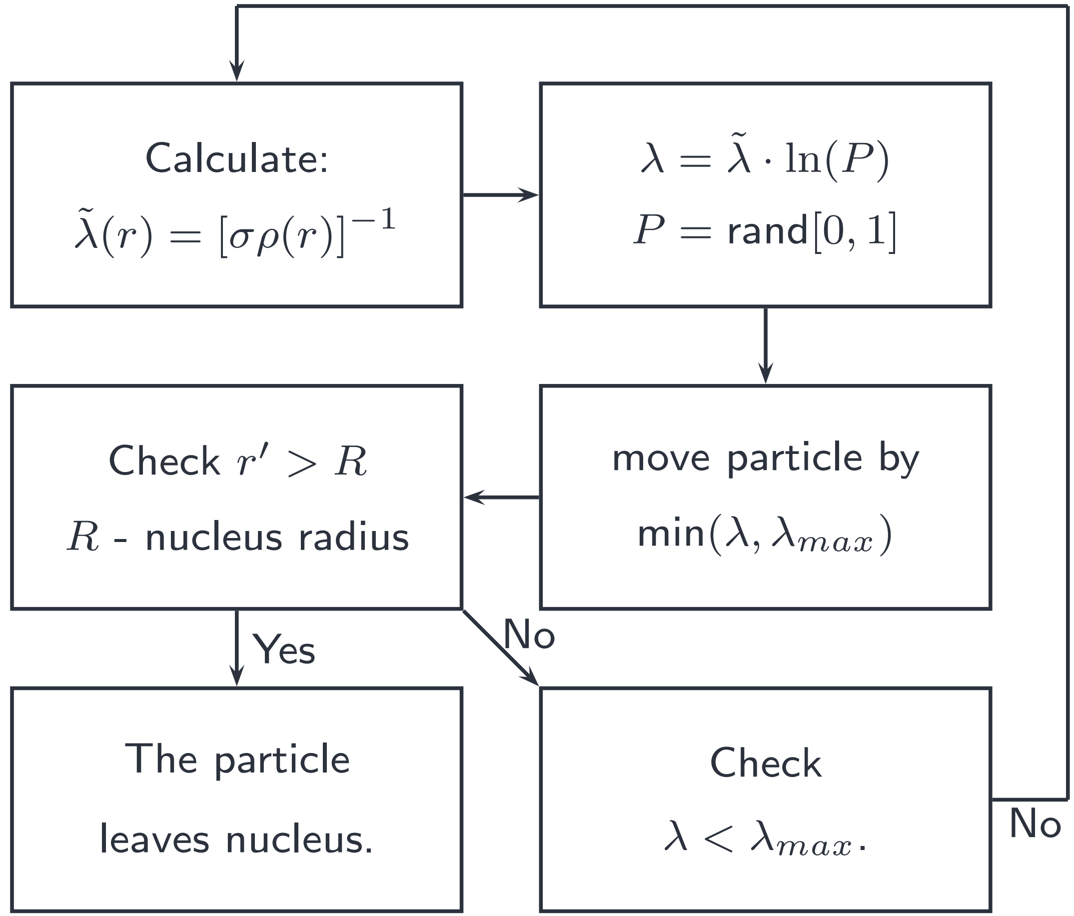
Check $r' > R$
R - nucleus radius

↓ Yes

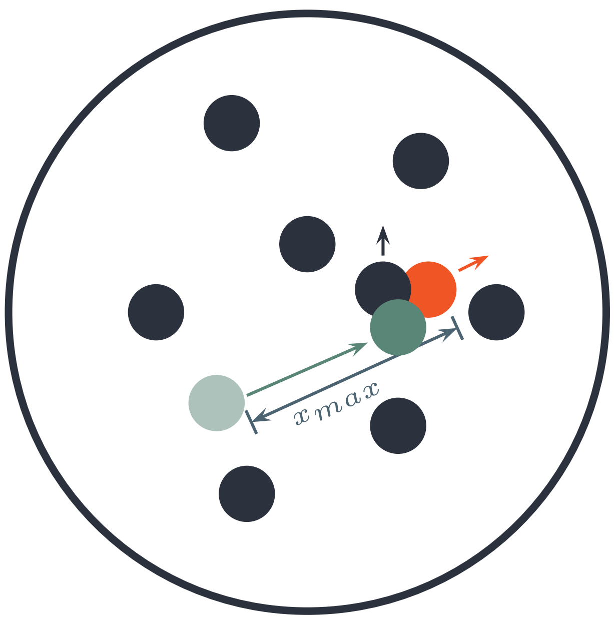
The particle
leaves nucleus.

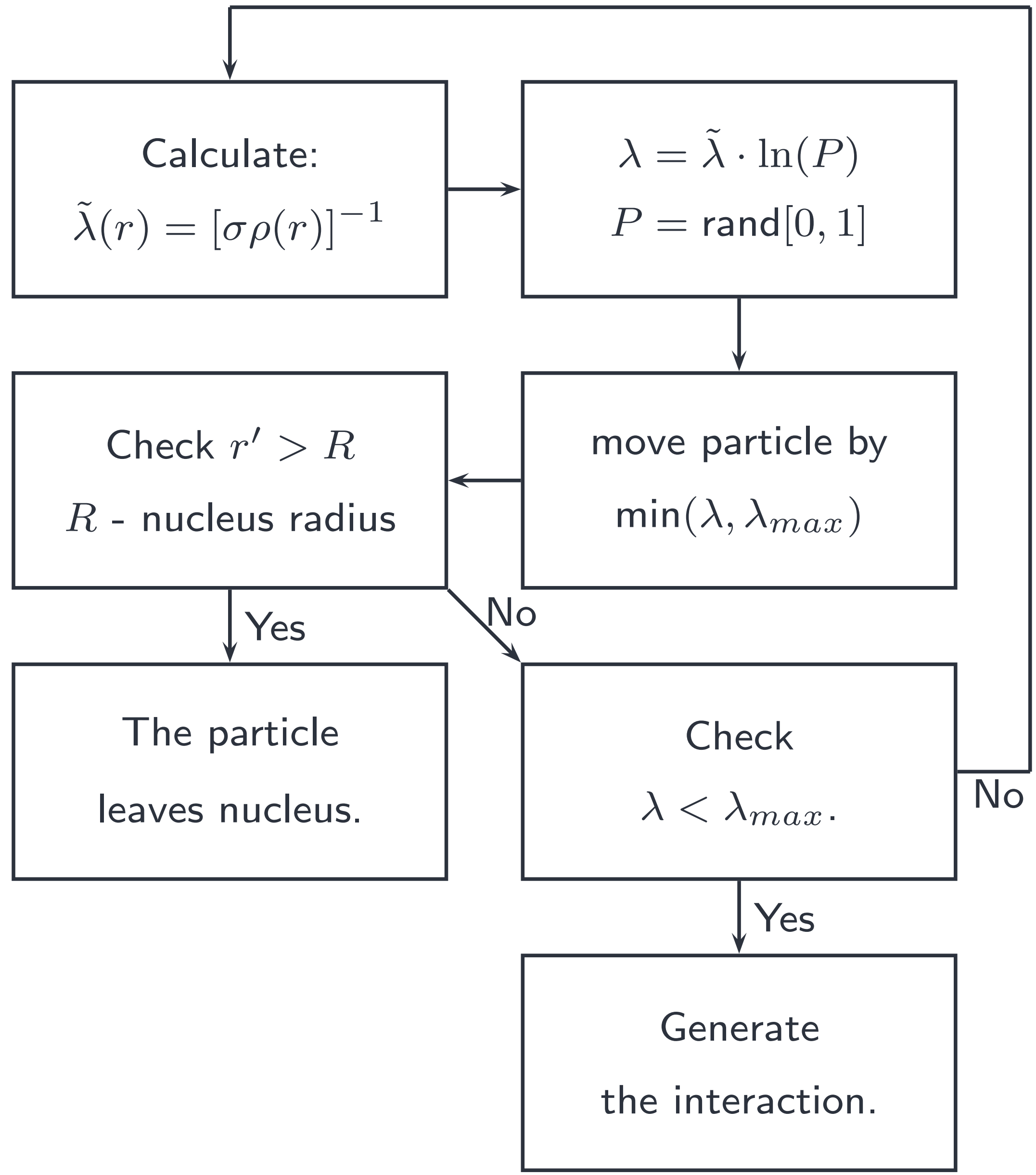
The algorithm for intranuclear cascade



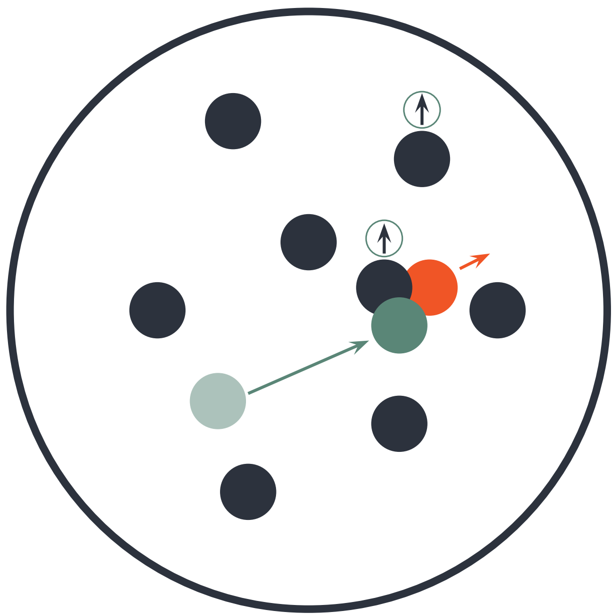


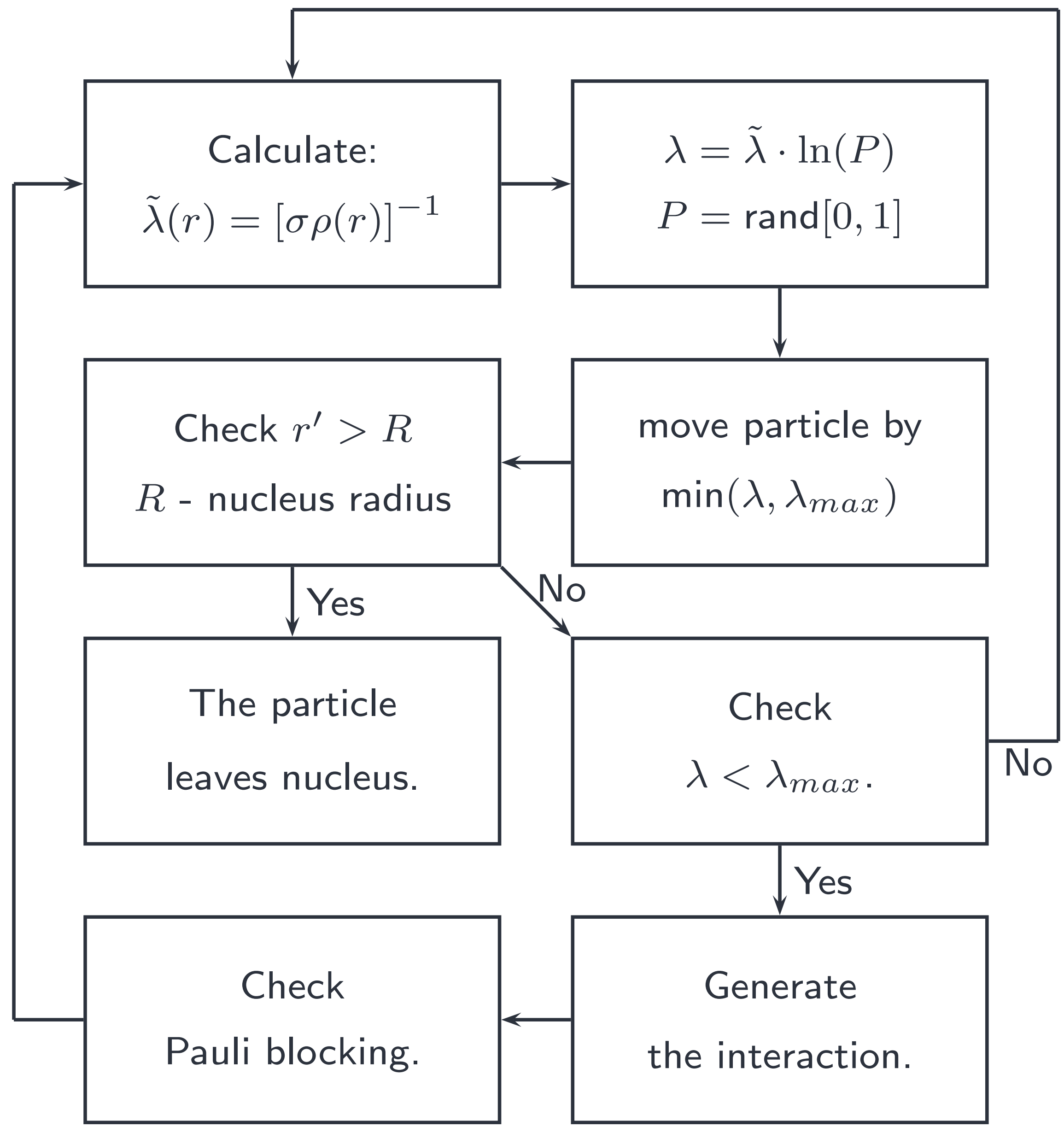
The algorithm for intranuclear cascade





The algorithm for intranuclear cascade





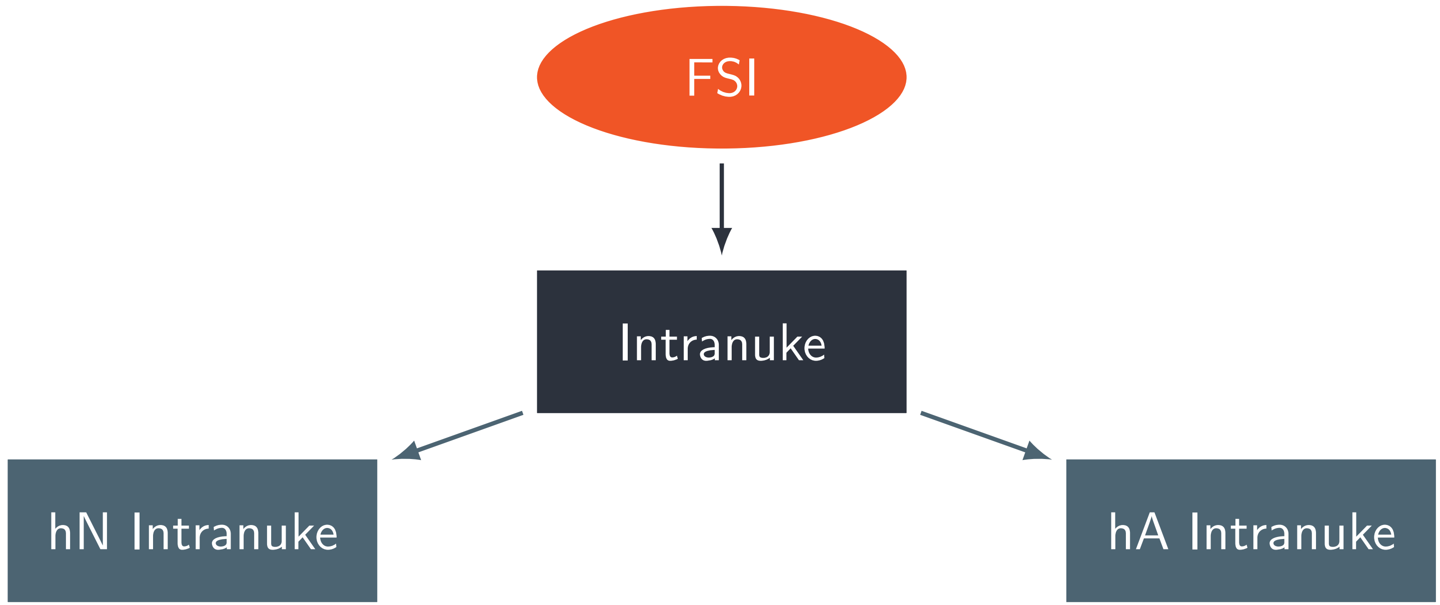
INC input

- The main input to the INC model is the particle-nucleon cross section
- Total cross section affects the mean free path
- Ratios of cross sections

$$\frac{\sigma_{qel}}{\sigma_{total}}, \quad \frac{\sigma_{cex}}{\sigma_{total}}, \quad \frac{\sigma_{abs}}{\sigma_{total}}, \quad \ldots$$

  are used to determine what kind of scattering happened
- NuWro and Neut use Oset model for low-energy pions and data-driven cross sections for all other cases
- GENIE has two models of FSI

FSI in GENIE



- intranuclear cascade
- data-driven cross sections
- Oset model for pions (coming soon)

- INC-like with one “effective” interaction
- tuned do hadron-nucleus data
- easy to reweight

Formation time

Landau Pomeranchuk effect

- The concept of formation time was introduced by Landau and Pomeranchuk in the context of electrons passing through a layer of material.

- For high energy electrons they observed less radiated energy then expected.

- The energy radiated in such process is given by:

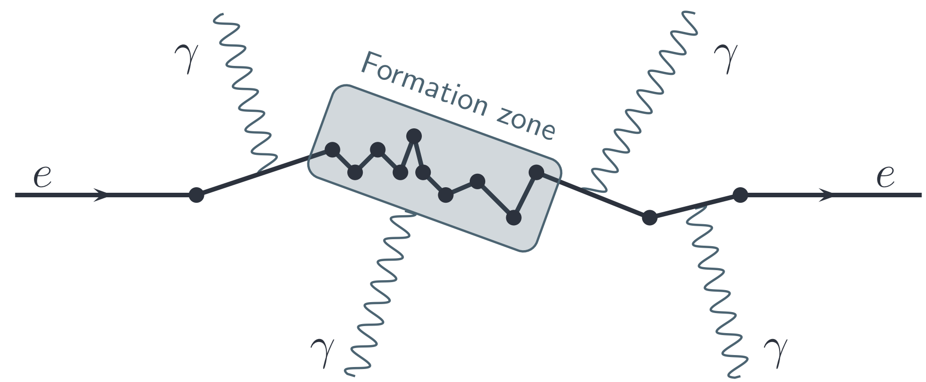
$$\frac{\mathrm{d}I}{\mathrm{d}^3k} \sim \left| \int_{-\infty}^{\infty} \vec{j}(\vec{x}, t) e^{i(\omega t - \vec{k}\cdot\vec{x}(t))} \mathrm{d}^3x \mathrm{d}t \right|^2$$

$\vec{x}(t)$ describes the trajectory of the electron.

ω, $\vec{k}$ are energy and momentum of the emitted photon.

- Assuming the trajectory to be a series of straight lines (the current density $j \sim \delta^3(\vec{x} - \vec{v}t)$) the radiation integral is:

$$\sim \int_{path} e^{i\left(\vec{k}\vec{v}-\omega\right)t} \mathrm{d}t$$



- Formation time is defined as:

$$t_f \equiv \frac{1}{\omega - \vec{k}\vec{v}} = \frac{E}{kp} = \frac{E}{m_e}\frac{1}{\omega_{r.f.}} = \gamma T_{r.f.}$$

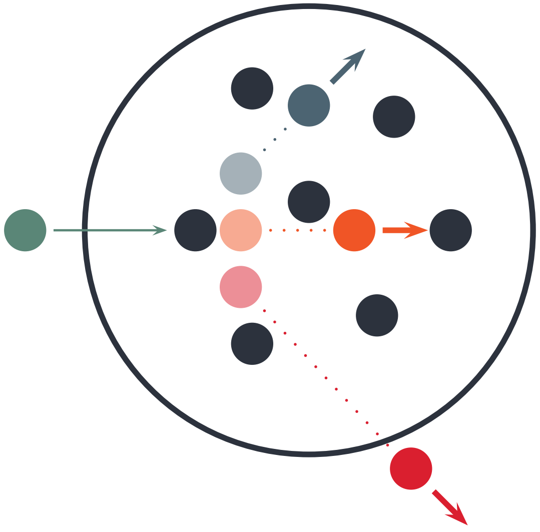
k, p - photon, electron four-momenta
$\omega_{r.f.}$ - photon frequency in the rest frame of the electron

- Formation time can be interpreted as the "birth time" of photon.

- If time between collisions $t >> t_f$, there is no interference and total radiated energy is just the average emitted in one collision multiplied by the number of collisions.

- If $t << t_f$, a photon is produced coherently over entire length of formation zone, which reduces the bremsstrahlung.

- One may expect a similar effect in hadron-nucleus scattering.
- In terms of INC it means that particles produced in primary vertex travel some distance, before they can interact.
- There are several parametrization used in MC generators
- Ranft parametrization:



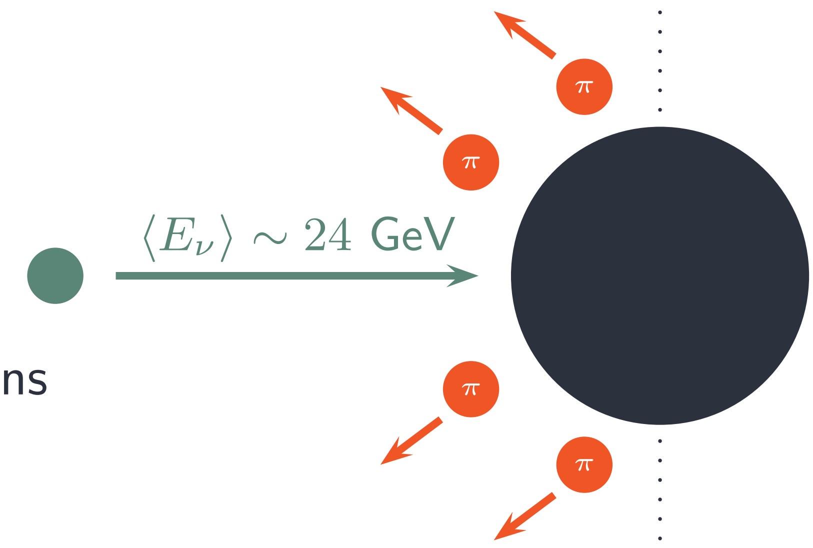
$$t_f = \tau_0 \frac{E \cdot M}{\mu_T^2}$$

where E, M - nucleon energy and mass, $\mu_T^2 = M^2 + p_T^2$ - transverse mass

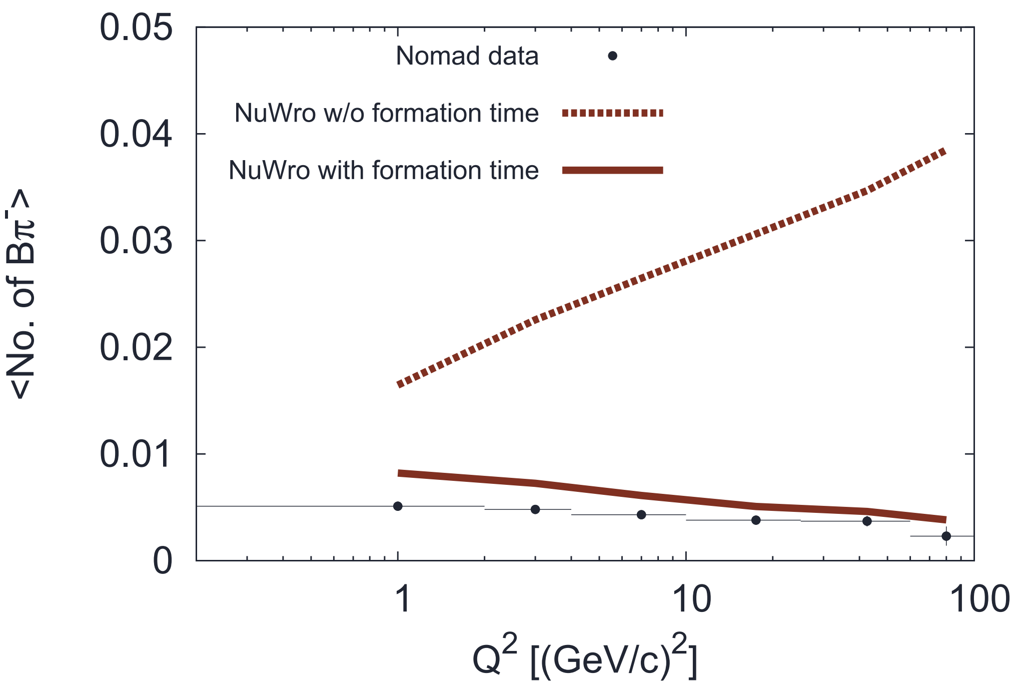
- SKAT parametrization (similar but with $p_T = 0$)
- NEUT and GENIE use SKAT parametrization
- NuWro uses Ranft parametrization for DIS and a model based on Δ lifetime for RES

Comparison with NOMAD data

- Nomad data from Nucl. Phys. B609 (2001) 255.
- The average number of backward going negative pions with the momentum from 350 to 800 MeV/c.



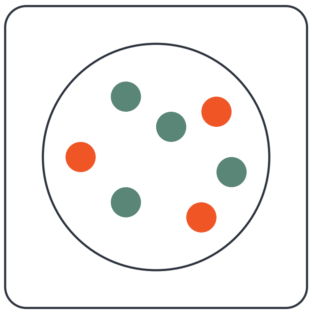
- In this neutrino energy range $B\pi^-$ are an effect of FSI.
- The observable is very sensitive to formation time effect.

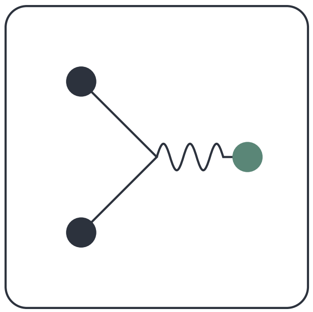


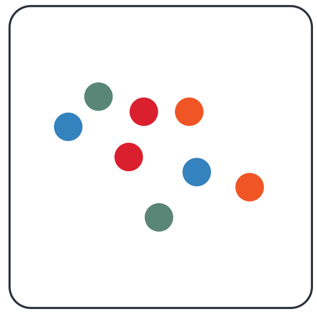
Summary

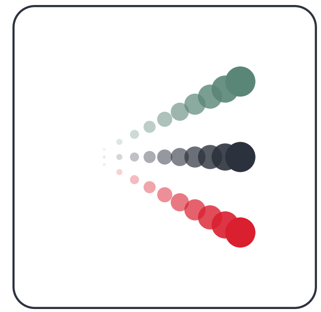
Neutrino-nucleus interactions

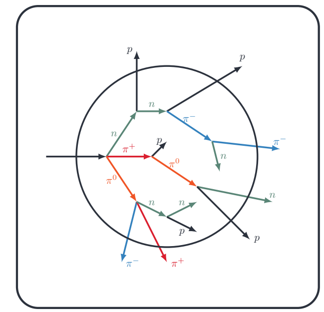
For all channels (but coherent) neutrino interactions are factorized in the following way


IA


νN


hadronization

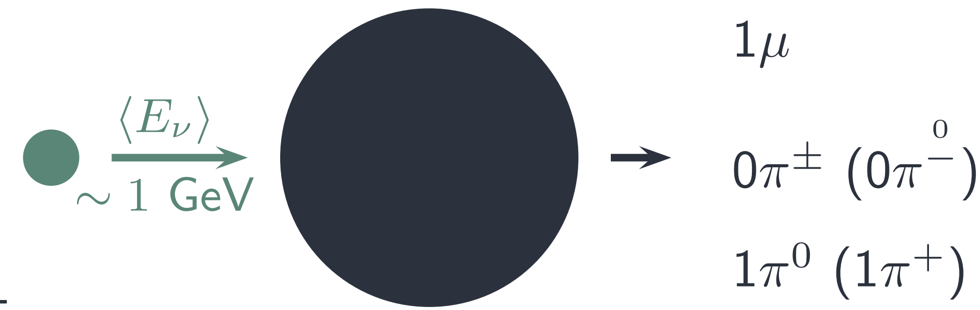

formation time


FSI

- Is the physics really factorized this way?
- This factorization is common for all generators
- However, some pieces are done in different way

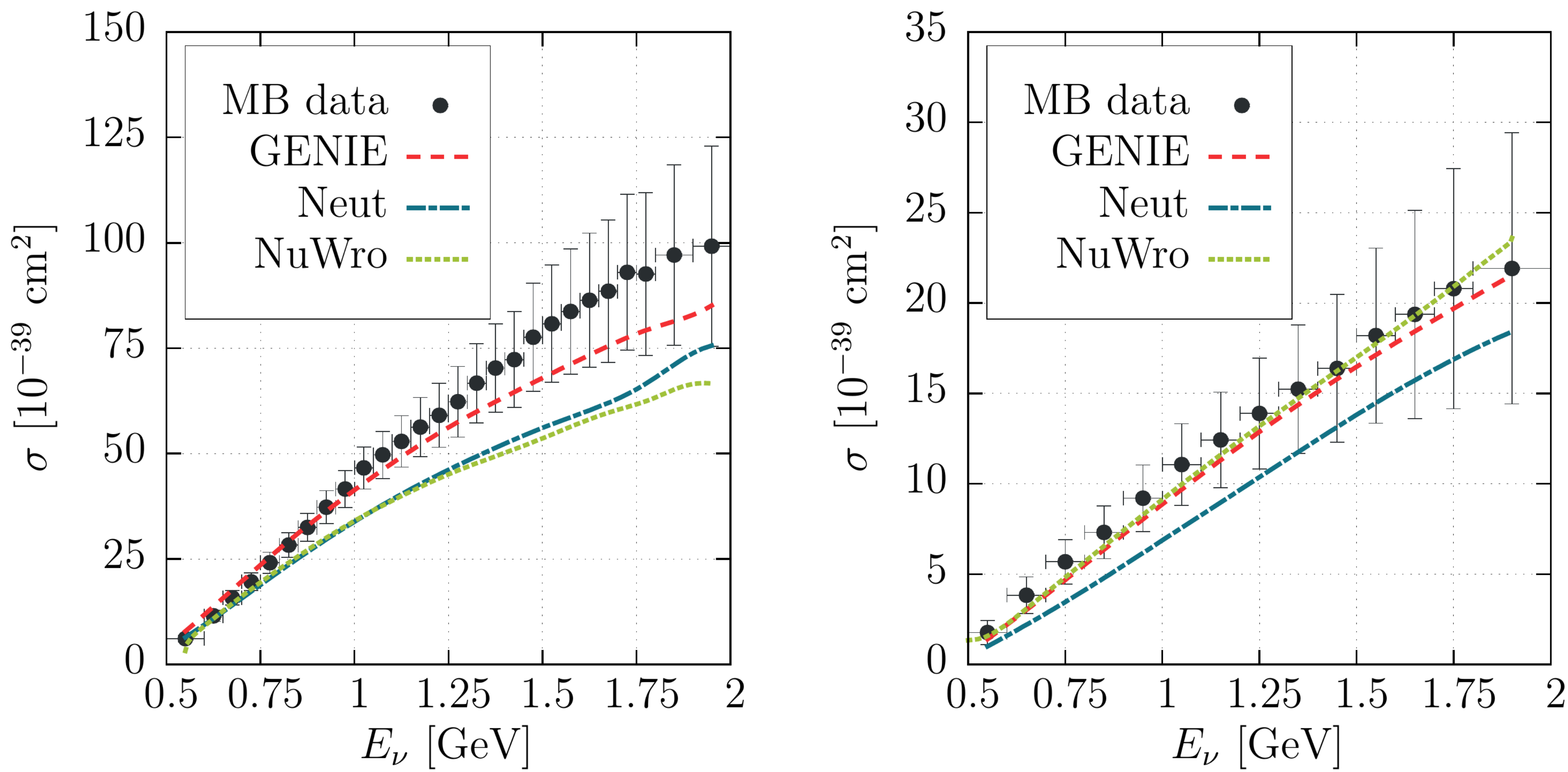
MiniBooNE data for CC π production

- The cross section for π^0 (π^+) production through charge current measured by MiniBooNE



- The signal is defined as: charged leptons, no charged pions and one neutral pion (one positive pion and no other pions) in the final state.

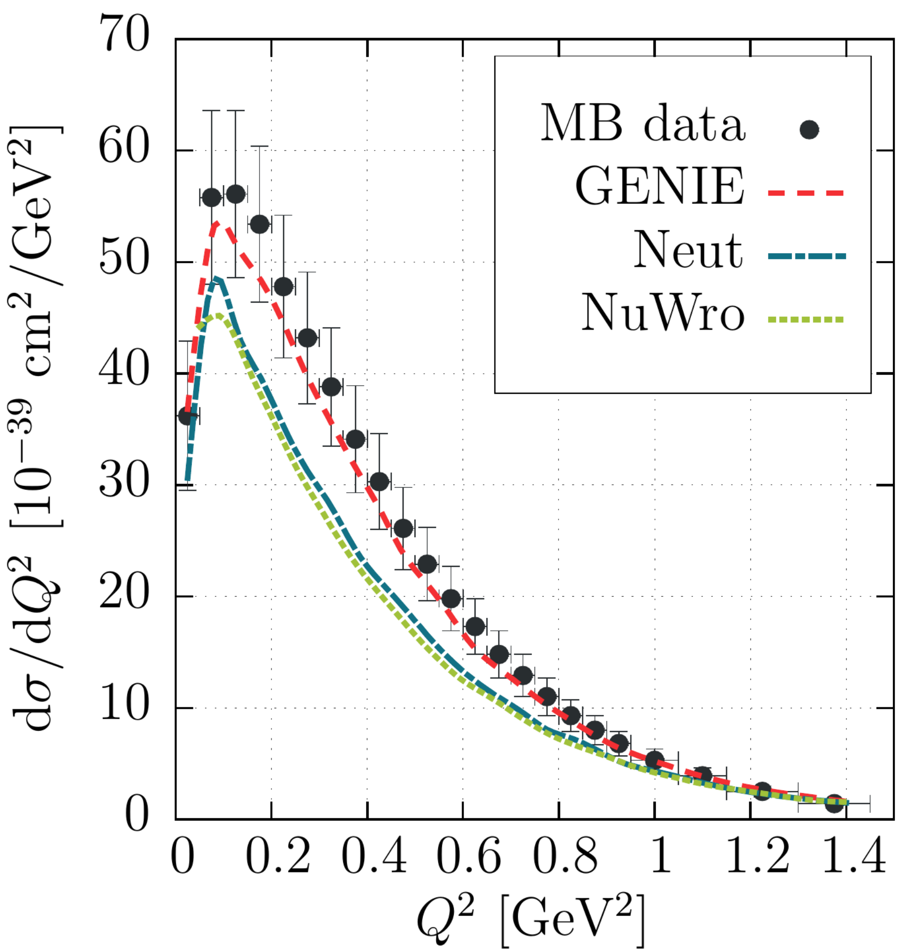
- The result depends on primary vertex and FSI, as pion can be:
  - produced in primary vertex
  - produced in FSI
  - affected by charge exchange
  - absorbed



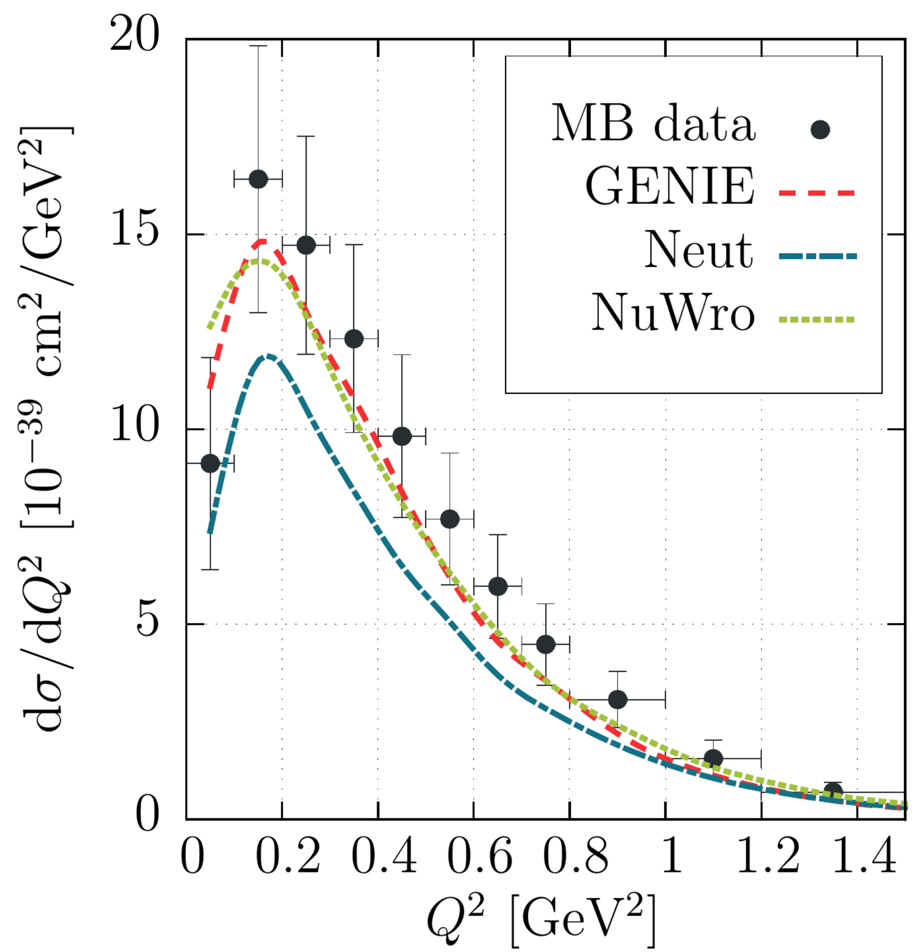
(a) $1\pi^+$ production

(b) $1\pi^0$ production

Figure 3.1: The total CC cross section for single pion production.



(a) $1\pi^+$ production

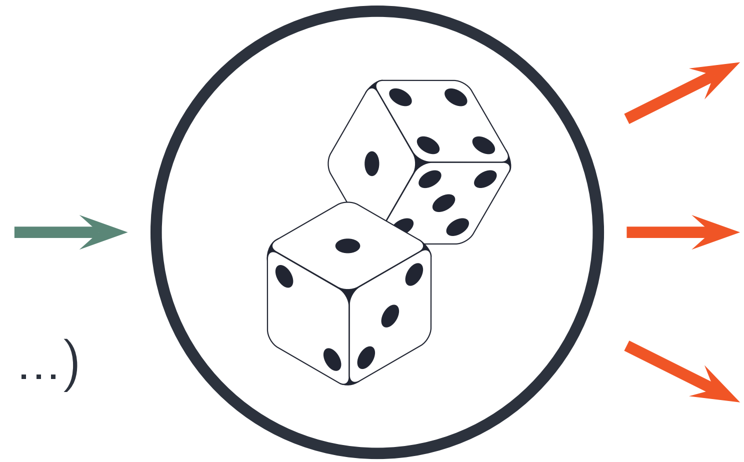


(b) $1\pi^0$ production

Figure 3.2: The differential CC cross section over Q^2 for single pion production.

Summary

- MC generators are irreplaceable tools in high-energy physics
- People use them before experiment exists (feasibility studies, requirements ...)



- And during data analysis (systematics uncertainties, backgrounds ...)
- There are several neutrino event generators and they all differ slightly
- And, unfortunately, there is no one right generator

Tutorial: analyzing MC output

- Each neutrino MC event generator performs simulations in two steps:
  - cross section calculation
  - event generation
- Usually, two output files are produced:
  - cross section per channel (sometimes as a function of energy - GENIE, sometimes integrated over flux - NuWro)
  - ROOT file with TTree of events

Introduction

- During tutorial we are going to work with GENIE's gst file (as it requires only ROOT)
- You can find it here: [gst file]
- If you are interested in more technical details on running generators visit the web page of NuSTEC school in Liverpool:

NuSTEC Neutrino Generator School

gst files

- The 'gst' is a GENIE summary ntuple format
- Selected branches:

iev	number of events
Q2	momentum transfer squared
W	invariant mass
nf	number of final state particles in hadronic system
nfpip	number of final state π^+
Ef(i)	energy of i-th particle in hadronic system

- You can find all of them in GENIE manual (p. 112): [GENIE manual]

Interactive ROOT

 # load ROOT file

```
root [0] TFile *myFile = new TFile ("file.gst.root")
```

 # load TTree

```
root [1] TTree *myTree = myFile->Get("gst")
```

 # lepton energy distribution

```
root [2] myTree->Draw("El")
```

 # lepton energy distribution for events with 1 hadron in the final state

```
root [3] myTree->Draw("El", "nf == 1")
```

 # use TBrowser if you prefer mouse over keyboard

```
root [4] TBrowser t
```

Script example

```
void analyzeGST (const char *inputFile)
{
  TFile *file = new TFile (inputFile);      // load root file
  TTree *tree = (TTree*)file->Get ("gst"); // get proper tree

  TH1D *leadingPip = new TH1D ("pip energy", "pip energy", 50, 0, 1);

  double leptonEnergy;        // lepton energy
  int nParticlesFS;           // number of particle in final state
  int nPip;                   // number of positive pion
  int hadronPDG[100];         // final state hadron pdg
  double hadronEnergy[100];   // final state hadron energy

  // set up branches
  tree->SetBranchAddress ("El",    &leptonEnergy);
  tree->SetBranchAddress ("nf",    &nParticlesFS);
  tree->SetBranchAddress ("nfpip", &nPip);
  tree->SetBranchAddress ("pdgf",  hadronPDG);
  tree->SetBranchAddress ("Ef",    hadronEnergy);
  ...
```

Script example

```
...
const int nEvents = tree->GetEntries(); // get number of events

for (int i = 0; i < nEvents; i++) // events loop
{
    tree->GetEntry (i); // get i-th event

    if (nPip == 0) continue;

    double maxEnergy = 0.0;

    for (int j = 0; j < nParticlesFS; j++) // particle loop
      if (hadronPDG[j] == 211 && hadronEnergy[j] > maxEnergy)
        maxEnergy = hadronEnergy[j];

    leadingPip->Fill (maxEnergy);
} // events loop

leadingPip->Draw();
}
```

Task 1: basic informations

- Using interactive ROOT find the following information on the simulation:
  - number of events
  - neutrino beam (flavor, energy)
  - target
  - dynamics

Task 2: basic distributions

- Using interactive ROOT plot the following distributions:
  - proton energy (before FSI)
  - proton energy (after FSI)
  - lepton energy
  - lepton energy for QEL
  - lepton energy for all events with no meson in the final state

Task 3: QEL

- Lets define QEL-like events as those without mesons in the final state
- On the same plot draw $\frac{d\sigma}{dQ^2}$ [in arbitrary units]:
  - total QEL-like
  - contribution from true QEL
  - background for QEL
- What cut could you apply to reduce background?

Task 4: reconstructed energy

- For QEL neutrino scattering on a nucleon at rest the incoming neutrino energy can be expressed by lepton kinematics:

$$E_\nu^{rec} = \frac{2(M_N - E_B)E_\mu - (E_B^2 - 2M_N E_B + m_\mu^2)}{2[M_N - E_B - E_\mu + |\vec{k}_\mu| \cos\theta_\mu]}$$

- Compare true and reconstructed neutrino energy for QEL events
- Compare true and reconstructed neutrino energy for QEL-like events