technische universität
dortmund

Statistical Methods of Data Analysis

Random Number Generation

Prof. Dr. Dr. Wolfgang Rhode Dr. Maximilian Linhoff

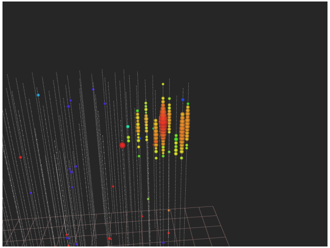
2023

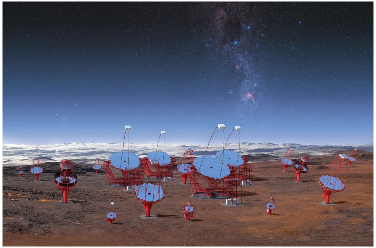
Overview

Overview

Occurrences of random numbers

- Random numbers can be found everywhere
- Everyday occurrences: Dice, shuffling cards, ...
- Physics: Particle interactions, simulations, ...




[1]

[1] CTA/M-A. Besel/IAC (G.P. Diaz)/ESO, Wikimedia

Pseudorandom number generators

- Why no “real” generators? (Atmospheric noise, electronic noise, ...)
  - Reproducibility
  - Searching for errors
  - Speed
- General rule:

$$\rightarrow x_{j+1} = f(x_1, \ldots, x_j)$$

  All generated random numbers are then completely deterministic!

Overview

Linear Congruent Generators (LCG)

Seed (“Start-value”): $x_0 \in \mathbb{N}_+$

$$x_{j+1} = ((a \cdot x_j + c) \bmod m)$$

$$\rightarrow u_j = x_j / m$$

- Example: $c = 3, a = 5, m = 16, x_0 = 0$

→ 0, 3, 2, 13, 4, 7, 6, 1, 8, 11, 10, 5, 12, 15, 14, 9, 0, ...

- Length of repeating sequence is referred to as *period length*
  - For LCGs the period length depends on a, c and m
  - An upper limit is given by m

Linear Congruent Generators (LCG)

$$\text{Seed ("Start-value"): } x_0 \in \mathbb{N}_+$$

$$x_{j+1} = ((a \cdot x_j + c) \bmod m)$$

$$\rightarrow u_j = x_j / m$$

- How to choose the parameters to obtain the maximal period length?
  1. $c \neq 0$
  2. c and m are coprime
  3. Each prime factor of m divides $(a - 1)$
  4. If m is divisible by 4, so is $(a - 1)$

XOR-Shift Generator

$$
\begin{aligned}
x_0 &\in \mathbb{N}_+ \\
t_1 &= \left((x_j \ll a) \oplus x_j\right) \\
t_2 &= \left((t_1 \gg b) \oplus t_1\right) \\
x_{j+1} &= \left((t_2 \ll c) \oplus t_2\right)
\end{aligned}
$$

- Period length is bound by k
  - Maximum period length: $2^k - 1$
- How to choose a, b and c?
  See George Marsaglia. "Xorshift RNGs". In: *Journal of Statistical Software* 8.14 (), pp. 1–6. DOI: `10.18637/jss.v008.i14`

Bitwise Operations:

$$
\begin{aligned}
k &: \text{Number of bits} \\
\gg &: \text{Bitwise shift} \\
\oplus &: \text{XOR Operation} \\
\oplus(0,0) &= 0 \\
\oplus(1,1) &= 0 \\
\oplus(1,0) &= 1 = \oplus(0,1) \\
x_j &= 11011100 \\
x_j \gg 3 &= 00011011 \\
(x_j \gg 3) \oplus x_j &= 11000111
\end{aligned}
$$

XOR-Shift Generator - Implementation

```python
def xorshift32(seed, N):
    '''
    Implementation of the "favorite choice" for 32-Bit from
    "George Marsaglia, Xorshift RNGs, Journal of Statistical Software".

    Every step performs a bit shift, XOR-operation (^ in python)
    and limits the result to the valid 32-bit range.
    '''
    x = seed
    a = 13
    b = 17
    c = 5
    for _ in range(N):
        x = ((x << a) ^ x) & 0xFFFF_FFFF
        x = ((x >> b) ^ x) & 0xFFFF_FFFF
        x = ((x << c) ^ x) & 0xFFFF_FFFF
        yield x
```

Mersenne-Twister

- Published in 1997
- One of the most used random number generators
    - Default in `numpy` until version 1.16, `C++11` standard library, `Root::TRandom3`
- Uses XOR-Shift amongst others to achieve bitwise random numbers
- Needs **624** Variables to save its state
- Generates **624** random numbers in each step
- Has a period length of

$$2^{19937} - 1$$

- Sounds great, but:
    - Also has some weaknesses in some statistical tests
    - Not very fast despite improvements to earlier iterations

Permutated Congruential Generators (PCG)

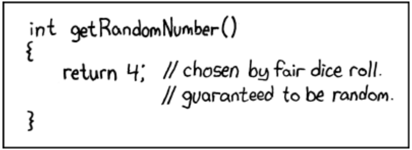
- Published 2014
- New standard in **`numpy`** since version 1.17
- Uses a LCG for its state
- Additionally uses bit-shifting transformation functions
- “extremely fast, extremely statistically good, and extremely space efficient”

See Melissa E. O’Neill. *PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation*. Tech. rep. HMC-CS-2014-0905. Claremont, CA: Harvey Mudd College, 2014. URL: `https://www.cs.hmc.edu/tr/hmc-cs-2014-0905.pdf`.

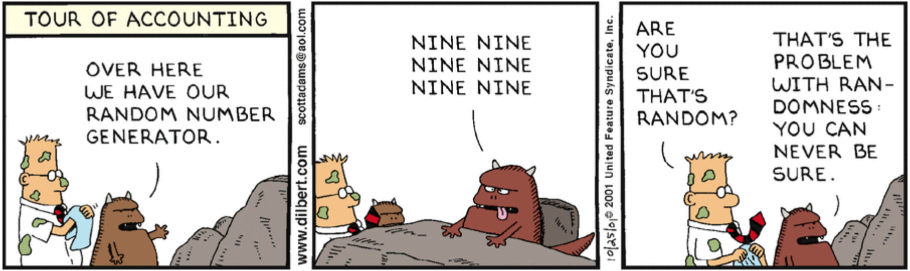
Overview

What is randomness?

- It is hard to “prove” randomness
- One can look for patterns in the generated numbers



[1]



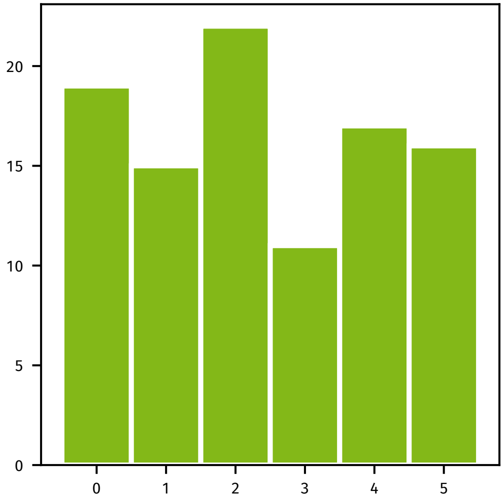
[2]

[1] https://xkcd.com/221/

[2] https://dilbert.com/strip/2001-10-25

Spectral test - 1D

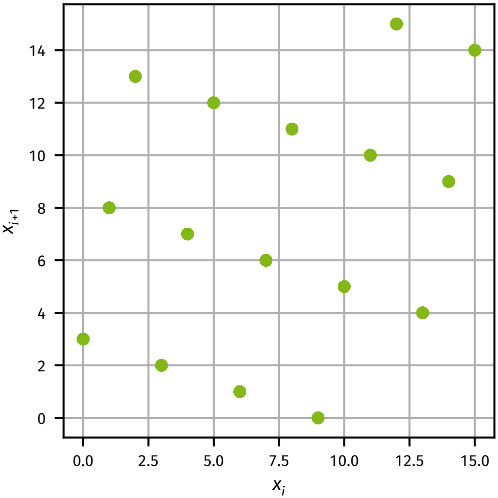
- Generally: How frequent do n-tuples of generated numbers occur?
- 1D: How frequent are numbers?
- Problem:
    - Disregards order of numbers
    - Example:
      0, 1, 2, 3, 5, 0, 1, 2, 3, 4, 0, 1, 2, 4, 5, ...



Spectral test - 2D

- 2D: How often do value-pairs occur?
- Example: **LCG** ($c = 3, a = 5, m = 16, x_0 = 0$)

→ $0, 3, 2, 13, 4, 7, 6, 1, 8, 11, 10, 5, 12, \ldots$



Spectral test - Procedure (2D)

1. Draw n random numbers (integers)
2. n^2 possible value pairs, only n value pairs are realized
3. Normalizing the integers results in a grid spacing of $1/m$
4. Finitely number of families (“parallel lines”) of straight lines can be laid through the occupied points
5. Consider the largest distance of adjacent families
6. If the lattice is **equally populated**, the distance of the line pairs is minimal: $d_2 = m^{-1/2}$
7. If the lattice is **unevenly populated**, the distance is $d_2 \gg m^{-1/2}$

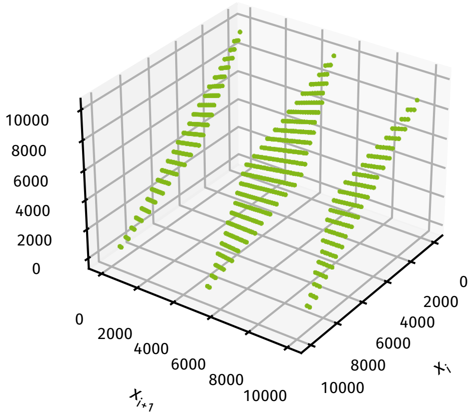
Spectral test - 3D

- 3D: How often do triplets occur?

→ Planes instead of lines

- Example:

$$\mathrm{LCG}(c = 3456, a = 1601, m = 10000, x_0 = 0)$$



Spectral Test - Higher Dimensions

- 4-dim? n-dim? Badly representable!
- (n-1)-dimensional hyperplanes
- For the n-dimensional case it follows:
  - If the grid is equally populated, the distance of the line pairs is the minimum realized distance $d_n \approx m^{-1/n}$
  - If the lattice is unevenly populated, the distance is $d_n \gg m^{-1/n}$

Other tests

- Birthday Spacing test
  - m birthdays in a year with n days
  - Distribution of the distances of all birthdays to each other should be Poisson distributed
  - In the actual test n and m are chosen much larger than in the classical problem
- Runs Test
  - Count number n of consecutive 0's or 1's in the generated numbers
  - Number n should be binomially distributed with $B(n, 0.5)$
- Testsuites
  - `Diehard`-Testsuite
  - `TestU01`

Hints for practical use of Pseudorandom Number Generators (PRNGs)

- PRNGs should generate the same sequence of numbers on all systems (Mostly not the case!)
- Seed: Wrongly chosen start parameters can shorten the period duration shorten strongly
- “Start-up time”: With some generators some of the first random numbers generated random numbers have to be discarded if the start parameters are start parameters are badly chosen (MT19937)
- Combine: If the period of a used PRNG is too short, several generators can be combined

Overview

Transformation of the uniform distribution

- Wanted: random variable $\boldsymbol{y}$ with the probability density

$$g(y); \quad y \in [y_{\text{min}}, y_{\text{max}}]$$

- Given: equally distributed random variable $\boldsymbol{u}$ with the probability density

$$f(u) = U(0,1) = \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \text{else} \end{cases}$$

- Relation:

$$g(y) = \left|\frac{\mathrm{d}u}{\mathrm{d}y}\right| \cdot f(u)$$

$$g(y) = \frac{\mathrm{d}G(y)}{\mathrm{d}y} \Rightarrow \mathrm{d}G(y) = g(y)\,\mathrm{d}y = U(0,1)\,\mathrm{d}u$$

Integrate:

$$u = \int_0^u U(0,1)\,\mathrm{d}u' = \int_{y_{\min}}^{y} g(y')\,\mathrm{d}y' = G(y)$$

$$y = G^{-1}(u)$$

→ Random variable with the desired distribution can be constructed by applying G^{-1} to the equally distributed variable u

Example

- Generation of random numbers in the range $[0, \pi]$ following $f(y) = \sin(y)$
  1. Create PDF $g(y)$ by normalizing $f(y)$

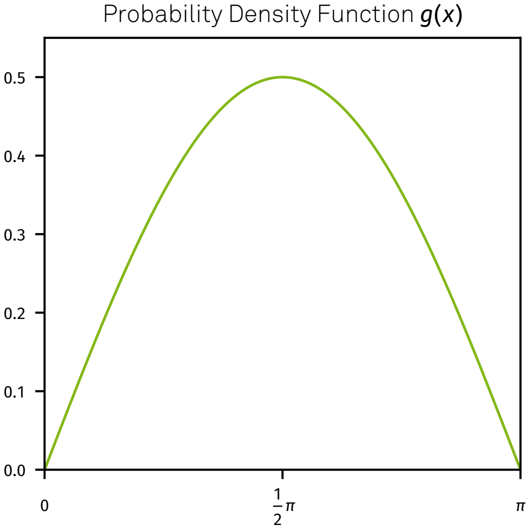
$$g(y) = \frac{\sin(y)}{\int_0^\pi \sin(y')\,\mathrm{d}y'} = \frac{1}{2}\sin(y)$$

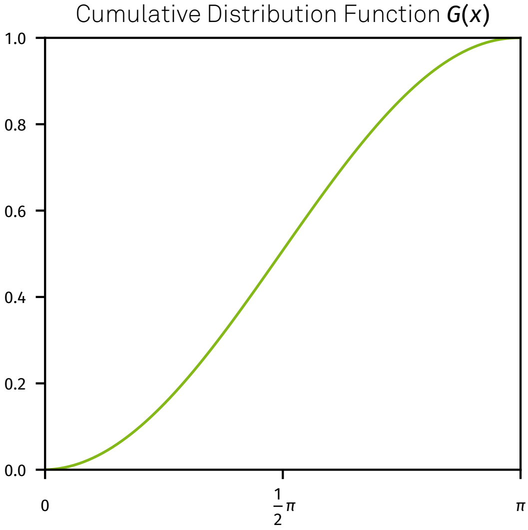
  2. Integrate up to y and invert

$$u = G(y) = \int_0^y \frac{1}{2}\sin(y')\,\mathrm{d}y' = \frac{1 - \cos(y)}{2}$$

$$y = G^{-1}(u) = \arccos(1 - 2u)$$

Example: sin(x)





- Advantages:
  - Very efficient
  - No need to discard drawn numbers
  - → No waste of computing time
- Disadvantages:
  - Only possible if the desired distribution is integrable
  - Inverse function must exist
- Alternative: Rejection sampling

Overview

Rejection Sampling (von Neumann)

- Wanted: Random variable $\boldsymbol{y}$ with probability density function

$$g(y); \quad y \in [y_{\text{min}}, y_{\text{max}}]$$

- $g(y)$ is not integrable or $G(y)$ is not invertable
- Given: Equally distributed random variables (u_1, u_2) with probability density functions:

$$f(u_1) = U(y_{\text{min}}, y_{\text{max}}) = \begin{cases} \frac{1}{y_{\text{min}} - y_{\text{max}}}, & y_{\text{min}} \leq x \leq y_{\text{max}} \\ 0, & \text{else} \end{cases}$$

$$f(u_2) = U(0, g_{\text{max}}) = \begin{cases} \frac{1}{g_{\text{max}}}, & g_{\text{max}} = \max(g(y)) \\ 0, & \text{else} \end{cases}$$

Rejection Sampling (von Neumann)

- Two random numbers must be drawn for each single, potential random number
- Many numbers are rejected (depending on the sampling pdf)

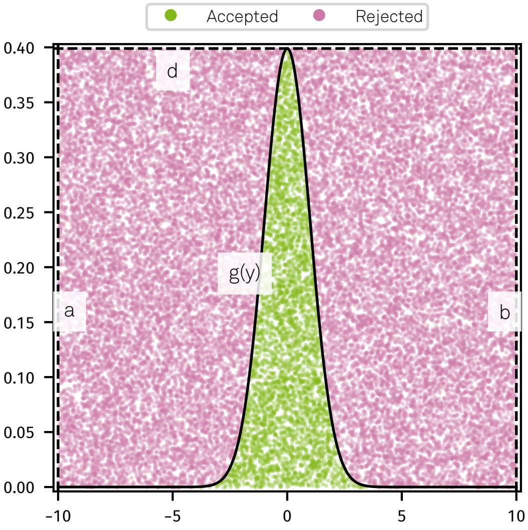
→ Inefficient ($E < 1$)

$$E = \frac{\int_a^b g(y)\,\mathrm{d}y}{(b-a)d}$$

$$\text{(If } g(y) \text{ is normed)} = \frac{1}{(b-a)d}$$

Rejection Sampling (von Neumann)

- Procedure:
  - If $g(u_1) \leq u_2$, reject u_1
  - If $g(u_1) > u_2$, accept u_1 as random number
- Example: Normal distribution between -10 and 10



Normal-distributed numbers: Box-Müller-method

- Issue: Normal distribution can not be integrated analytically

→ Can not use the transformation method

- Solution: Integrate in 2D

$$I^2 = \frac{1}{2\pi} \int_{-\infty}^{x} \mathrm{d}x' \int_{-\infty}^{y} \mathrm{d}y' \exp\left(-\frac{1}{2}\left(x'^2 + y'^2\right)\right)$$

Box-Müller-Method

- Transform to polar coordinates:

$$x = r\cos\phi$$
$$y = r\sin\phi$$
$$I^2 = \frac{1}{2\pi}\int_0^{2\pi} \mathrm{d}\phi \int_0^{r} \mathrm{d}r'\, r' \exp\left(-\frac{1}{2}r'^2\right)$$

→ Use the inversion method for ϕ and r

Box-Müller-Method

1. Inversion method for ϕ:

$$u_1 = F(\phi) = \frac{1}{2\pi} \int_0^{\phi} \mathrm{d}\phi \int_0^{\infty} \mathrm{d}r' \; r' \exp\left(-\frac{1}{2} r'^2\right) = \frac{\phi}{2\pi} \leftrightarrow \phi = 2\pi u_1$$

2. Inversion method for r:

$$u_2 = F(r) = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{d}\phi \int_0^{r} \mathrm{d}r' \; r' \exp\left(-\frac{1}{2} r'^2\right) = 1 - \exp\left(-\frac{1}{2} r^2\right) \leftrightarrow r = \sqrt{-2 \ln(u_2)}$$

Box-Müller-Method

- After transforming back, x and y are independent random variables:

$$x = r\cos(\phi) = \sqrt{-2\ln(u_2)}\cos(2\pi u_1)$$
$$y = r\sin(\phi) = \sqrt{-2\ln(u_2)}\sin(2\pi u_1)$$

- Polar-Method: Instead of the analytical calculation of the trigonometric function, use the rejection method

Polar method

- Create equally distributed u_1, u_2
- Transform: $v_1 = 2u_1 - 1$, $v_2 = 2u_2 - 1$
- Calculate $s = v_1^2 + v_2^2$, reject if $s \geq 1$
- Use as independent, normally distributed random numbers:

$$x_1 = v_1\sqrt{-\frac{2}{s}\ln(s)}$$

$$x_2 = v_2\sqrt{-\frac{2}{s}\ln(s)}$$

Polar method - Reasoning / Proof

- Polar coordinates of point (x_1, x_2):

$$x_1 = \cos\theta\sqrt{-2\ln(s)}$$
$$x_2 = \sin\theta\sqrt{-2\ln(s)}$$

- Distribution function for $\sqrt{-2\ln(s)} \leq r = \sqrt{s}$:

$$\begin{aligned} F(r) &= P\left(\sqrt{-2\ln(s)} \leq r\right) \\ &= P(-2\ln(s) \leq r^2) \\ &= P(s \geq e^{-r^2/2}) \end{aligned}$$

Polar method - Reasoning / Proof

- $s = r^2$ is equally distributed between $[0, 1]$

→ $F(r) = 1 - e^{-r^2/2}$

- Corresponding probability density:

$$f(r) = \frac{\mathrm{d}F(r)}{\mathrm{d}r} = re^{-r^2/2}$$

- Now create the combined distribution function

Polar method - Reasoning / Proof

- Common distribution function:

$$
\begin{aligned}
F(x_1, x_2) &= P(x_1 \leq k_1, x_2 \leq k_2) = P(r\cos(\theta) \leq k_1, r\sin(\theta) \leq k_2) \\
&= \frac{1}{2\pi} \int_{x_1 \leq k_1} \int_{x_2 \leq k_2} r e^{-r^2/2} \, \mathrm{d}r \, \mathrm{d}\phi \\
&= \frac{1}{2\pi} \int_{x_1 \leq k_1} \int_{x_2 \leq k_2} e^{-(x_1^2 + x_2^2)/2} \, \mathrm{d}x_1 \, \mathrm{d}x_2 \\
&= \left(\frac{1}{2\pi} \int_{-\infty}^{k_1} e^{-x_1^2/2} \, \mathrm{d}x_1 \right) \cdot \left(\frac{1}{2\pi} \int_{-\infty}^{k_2} e^{-x_2^2/2} \, \mathrm{d}x_2 \right) \\
&= \mathcal{N}(x_1) \cdot \mathcal{N}(x_2)
\end{aligned}
$$

Generation of Poisson-distributed random numbers

- Possibility 1:
  - Generate exponentially distributed random variables
  - Sum u_i until the sum exceeds μ (i_{max} numbers)
  - Random number: $x = i_{\mathrm{max}} - 1$
- Reasoning:
  - Logarithm of exponentially distributed values = equally distributed values

Reminder:

$$P(r) = \frac{\mu^r e^{-\mu}}{r!}$$

Generation of Poisson-distributed random numbers

$$\frac{1}{\tau}e^{-t/\tau} \text{ with } t = -\tau \ln(x)$$

$$\rightarrow \sum_i t_i = -\tau \sum_i \ln(x_i)$$

$$\rightarrow \frac{\sum_i t_i}{\tau} = -\sum_i \ln(x_i)$$

$$\rightarrow e^{-\frac{\sum_i t_i}{\tau}} = \prod_i x_i$$

Generation of Poisson-distributed random numbers

- Possibility 2:
  - For large μ: Poissonian ≈ Gaussian
  - “Large”: $\mu > 10$
- For a normally distributed random variable Z:

$$n = \max(0, \mathrm{int}(\mu + Z\sqrt{\mu} + 0.5))$$

Generation of χ^2-distributed random numbers

- For even n:
  - Construct product of $n/2$ equally distributed random numbers

$$x = -2 \ln \left(\prod_{i=1}^{n/2} u_i \right)$$

→ x is a χ^2-distributed random variable

Generation of χ^2-distributed random numbers

- For odd n:
  - Add the square of a normally distributed random variable Z

$$x = -2\ln\left(\prod_{i=1}^{n/2} u_i\right) + Z^2$$

→ x is a χ^2-distributed random variable

Generation of χ^2-distributed random numbers

- For large n:
  - Approximate with Gaussian distribution
  - The variable y is approximately normally distributed:

$$y = \sqrt{2\chi^2} - 2\sqrt{2n-1}$$

  - Generate a random number Z which is standard-normal distributed
  - Calculate

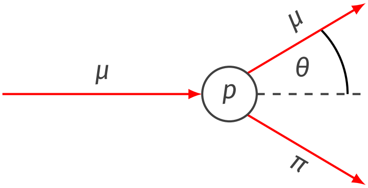
$$x = \frac{1}{2}\left(Z + \sqrt{2n-1}\right)^2$$

  - Reject if

$$Z < (-\sqrt{2n-1})$$

Applications in particle physics

- Examples:
  - Inelastic Muon-Nucleus scattering
  - π-production
  - Local reference frame



- Given: Double differential cross section:

$$\frac{d^2\,\sigma(E,\nu,\theta)}{d\theta\,d\nu}$$

Calculation

1. Calculate the total cross section:

$$\sigma_{\text{total}}(E) = \int_{v_{\min}}^{v_{\max}} \int_{\theta_{\min}}^{\theta_{\max}} \frac{\mathrm{d}^2\,\sigma(E,v,\theta)}{\mathrm{d}\theta\,\mathrm{d}v}\,\mathrm{d}v\,\mathrm{d}\theta$$

2. Calculate the (total) probability P_w for an interaction in a medium:

$$P_W = \frac{N_A \rho L}{A} \int_{E=0}^{E=\infty} \sigma(E) f(E)\,\mathrm{d}E$$

ρ = Density of medium

A = Atomic weight

L = Length of path

N_A = Avogadro constant

$f(E)$ = Probability of particle energy

- Caution:
  - If L is chosen too large, P will be greater than 1!

  → Wrong results!
  - If L is chosen very small, a lot of operations have to be performed until an interaction occurs

  → Waste of resources, numerical problems
- The probability P_E for n interactions on the path L is Poisson distributed:

$$P_e(n, \lambda = P_W) = \frac{\lambda^n}{n!} e^{-\lambda}$$

Variant a: Thin detector

- P_W is small
- Calculate the counter probability for zero interactions:

$$1 - P_E(n = 0)$$

- Now energy and location of the interaction are known

Variant b: Thick Detector

- Calculate the mean free path length:

$$l = \frac{\int x P_{E(n=0)}(x)\,\mathrm{d}x}{\int P_{E(n=0)}(x)\,\mathrm{d}x}$$

- Calculate the interaction probability as the counter probability for zero interactions:

$$P_{\text{int}} = 1 - P(x) = 1 - e^{-\frac{x}{l}}$$

- Calculate the interaction point with the transformation method