DEYDAA 2026

Monte Carlo Teknikleri

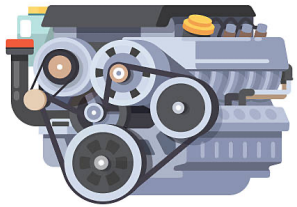
Rastgele Sayılardan Fiziksel Dağılımlara
Sinop Gerze Deydaa Parçacık Fiziği Yaz Okulu

Halil Gamsızkan
Eskişehir Teknik Üniversitesi
Fizik Bölümü

Üstten bakış

Dersteki konular bir zincir üzerinden ilerleyecek: integralden fiziksel yoruma.

1. Derste matematiksel altyapıyı kuracağız;



2. Derste bu altyapının çarpışma olayları üretimindeki kullanımını görecek ve MC olay üretim zincirini tanıyacağız.

integral → rastgele örnek → weight → histogram → belirsizlik → HEP event

Deneyde beklenen olay sayısı

Beklenen olay sayısı, tesir kesiti ile kabul/verimin çarpımından gelir.

σ: fizik (tesir kesiti);
$A \cdot \epsilon$: olay seçimi ve dedektör;
$\mathscr{L}$: deney (ışınlık).

Soru:

σ,
dağılımlar
ve MC örneklemleri (sample)

pratikte nasıl üretilir?

$$N = \mathcal{L}\,\sigma\,A\,\epsilon$$

Deneyde Ölçülen — Kuram Bilgisi

Neden N? Çünkü deneyde sonunda yaptığımız şey çoğu zaman olay saymak. Teori ile veriyi karşılaştırabilmek için, belirli bir süreçten ve belirli bir seçimden sonra kaç olay görmeyi beklediğimizi bilmek istiyoruz.

Faz Uzayı ve Tesir kesiti

Bir çarpışma olayının son durumundaki parçacıkların:

- enerjileri,
- momentumları,
- polar ve azimut açıları,
- rapiditeleri

vardır.

Bunların tamamına birlikte bir **faz uzayı noktası** diyelim: Φ

Teori bize kabaca $\frac{d\sigma}{d\Phi}$ verir.

Bu veri şu soruyu cevaplar:

> Faz uzayının Φ civarındaki küçük bir bölgesi tesir kesitine ne kadar katkı yapıyor?

Toplam tesir kesiti ise bütün izinli faz uzayı üzerinden:

$$\sigma = \int d\Phi \frac{d\sigma}{d\Phi}$$

olarak bulunur.

Faz Uzayı ve Tesir kesiti

Kuantum alan kuramı bize belirli bir başlangıç ve son durum konfigürasyonu için bir **genlik** verir:

$\mathscr{M}(\Phi)$.

Bunun karesi,

$|\mathscr{M}(\Phi)|^2$,

o konfigürasyonun göreli olarak ne kadar güçlü olduğunu belirler. Fakat toplam süreçte yalnızca tek bir son durum geometrisi yoktur.

Örneğin

$e^+e^- \to \mu^+\mu^-$

sürecinde müonlar farklı açılarda çıkabilir. Teori önce açısal dağılımı verir:

$$\frac{d\sigma}{d\cos\theta}.$$

Tüm açılardaki toplam kesiti bulmak için:

$$\sigma = \int_{-1}^{1} \frac{d\sigma}{d\cos\theta} d\cos\theta$$

hesaplanır.

Burada integralin fiziksel anlamı son derece somut:

> **Müonların çıkabileceği bütün açıların kesit katkılarını topluyoruz.**

Basit durumda analitik hesap mümkün

Örneğin ağaç seviyesinde

$$e^+e^- \to \mu^+\mu^-$$

için kütleleri ihmal edersek:

$$\frac{d\sigma}{d\cos\theta} \propto 1 + \cos^2\theta$$

gibi basit bir ifade elde edebiliriz. Sonra

$$\sigma = \int_{-1}^{1} \frac{d\sigma}{d\cos\theta} d\cos\theta$$

integralini **analitik olarak** alırız.

Burada:

- başlangıç durumu belirli,
- son durumda iki parçacık var,
- tek anlamlı değişken büyük ölçüde saçılma açısı,
- karmaşık seçimler yok.

Dolayısıyla toplam tesir kesitini bulmak için MC (Monte Carlo) veya başkaca bir nümerik tekniğe ihtiyacımız yok.

Proton çarpıştırıcısında daha da fazla integral var

LHC'de başlangıçta çarpışan temel nesneler **protonların** tamamı değil, proton içindeki **partonlardır**. Bu partonların proton momentumunun **hangi oranını** taşıdığı da olaydan olaya değişir: x_1, x_2.

Dolayısıyla hadron seviyesindeki kesitte şunları da toplamak gerekir:

$$\sigma_{pp\to X} = \sum_{a,b} \int dx_1\, dx_2\, f_a(x_1, \mu_F) f_b(x_2, \mu_F) \int d\Phi \frac{d\hat{\sigma}_{ab\to X}}{d\Phi}.$$

Burada:

- a, b: olası başlangıç parton türleri,
- x_1, x_2: partonların momentum oranları,
- f_a, f_b: PDF'ler,
- Φ: son durum kinematiği.

Yani şunların hepsini topluyoruz:

1. Hangi partonlar çarpıştı?
2. Momentum oranları neydi?
3. Son durum parçacıkları hangi kinematikle çıktı?

Bu yüzden integral **çok boyutlu** hale geliyor.

PDF'ler genellikle basit analitik fonksiyonlar değildir. Global fitlerden elde edilmiş, sayısal gridler ve interpolasyonlar olarak kullanılırlar. Bu nedenle (x_1, x_2) integralleri **pratikte nümerik yapılır**.

Son durum karmaşıklaştıkça faz uzayı büyür

Diyelim ki süreç:

$$pp \to Z + 2j \to \ell^+\ell^- + 2j\,.$$

Son durumda **dört parçacık** varsa, **her birinin** üç bağımsız momentum bileşeni vardır.
Enerji-momentum korunumu çıkarıldığında bile çok sayıda **bağımsız** değişken kalır.

Genel olarak n son durum parçacığı için faz uzayı yaklaşık $3n - 4$ boyutludur.

Üstelik buna:

- $x_1, x_2,$
- rezonans bozunmaları,
- parçacık kütleleri,
- spin korelasyonları,
- farklı partonik kanallar

eklenir.

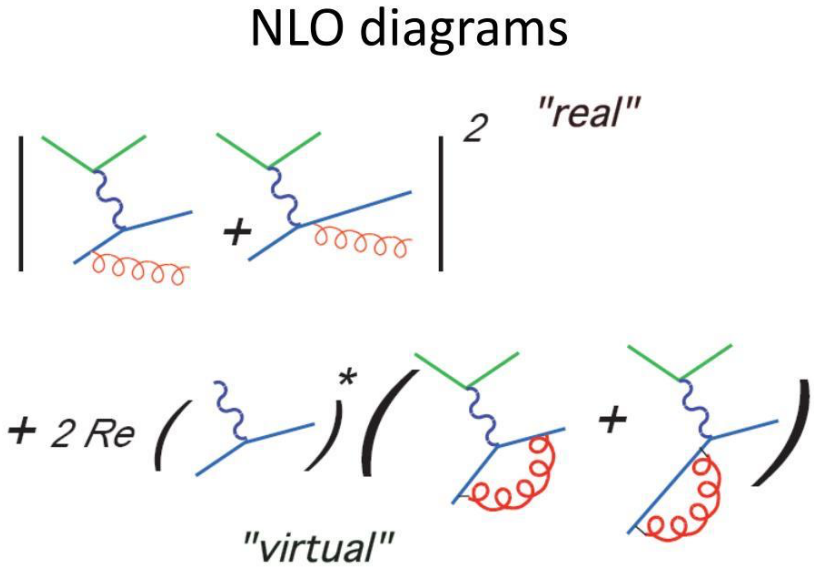
Yani integral birkaç boyutlu değil, rahatlıkla **10–30** veya **daha fazla** boyutlu olabilir.

Yüksek mertebelerde işler daha da zorlaşıyor

LO'da bile integral karmaşık olabilir. NLO'da ayrıca:

- sanal loop katkıları,
- bir ekstra parçacıklı gerçek radyasyon,
- infrared tekillikler,
- subtraction terimleri

vardır.



İntegraller için grid yöntemleri neden yetmiyor?

Mesela tek boyutta

$$I = \int_0^1 f(x)dx$$

hesaplamak istiyoruz. [0,1] aralığını 100 eşit parçaya böleriz:

$x_1, x_2, \ldots, x_{100}$.

Her noktada $f(x)$'i hesaplayıp Riemann toplamı, trapez yöntemi vb. kullanırız. Bu noktaların oluşturduğu düzenli yapıya **grid** diyoruz.

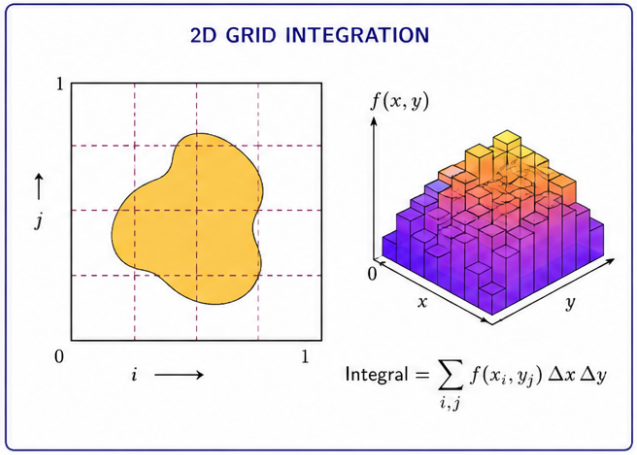
İki boyutta integral varsa,

$$I = \int f(x,y)\,dxdy$$

her eksende 100 nokta seçersek toplam:

$$100 \times 100 = 10^4$$

grid noktası gerekir.



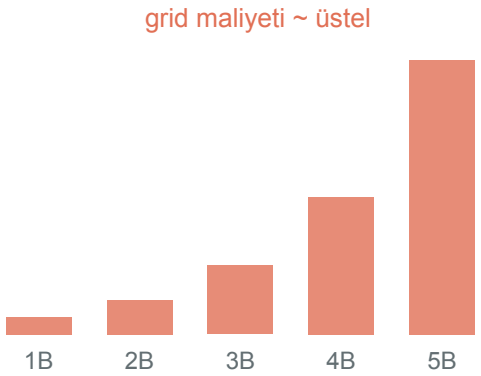
İntegraller için grid yöntemleri neden yetmiyor?

Üç boyutta:

$$100^3 = 10^6 .$$

10 boyutta:

$$100^{10} = 10^{20} .$$



Boyut laneti (curse of dimensionality): her boyuta makul sayıda nokta koysan bile toplam grid noktası sayısı **üstel şekilde patlıyor**.

HEP'te faz uzayı örneğin 15–20 boyutluysa, düzenli bir grid ile "her bölgeyi sistematik tarayayım" yaklaşımı pratik olmaktan çıkıyor.

Monte Carlo'nun avantajı şu: tüm grid noktalarını gezmek yerine faz uzayında **rastgele noktalar örnekliyoruz**. Hata da tipik olarak

$$\delta_{\mathrm{MC}} \sim \frac{1}{\sqrt{N}}$$

şeklinde azalıyor; bu üssün $(-1/2)$ boyutla doğrudan kötüleşmemesi yüksek boyutta MC'yi cazip hale getiriyor.

Monte Carlo fikri: integral = ortalama

$$I = \int_0^1 f(x)dx$$

ifadesini olasılık diliyle yazarsak, x değişkenini [0,1] aralığında düzgün (uniform) seçtiğimizde:

$x \sim U(0,1)$ ve yoğunluğu $p(x) = 1$ olur. Bu durumda $f(x)$'in beklenen değeri:

$$\mathbb{E}[f(X)] = \int_0^1 f(x)p(x)dx = \int_0^1 f(x)dx\,.$$

(Kuantum fiziğinden hatırlayalım: $\langle x \rangle = \int_{-\infty}^{\infty} x\,|\,\psi(x)\,|^2\,dx$)

Dolayısıyla:

$I = \mathbb{E}[f(X)]$=dağılımın kuramsal ortalaması

Monte Carlo fikri: integral = ortalama

Sonra beklenen değeri, rastgele çekilmiş örneklerin ortalamasıyla **tahmin ediyoruz**:

$$\mathbb{E}[f(X)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)\,.$$

Yani:

- Kuramsal beklenen değer: $\mathbb{E}[f(X)]$
- MC ile örneklem ortalaması: $\frac{1}{N} \sum_{i} f(x_i)$

Büyük sayılar yasası nedeniyle (N) arttıkça ikinci ifade birinciye yaklaşır.

Monte Carlo fikri: integral = ortalama

örnek:

$$f(x) = x^2 \qquad X \sim U(0,1)$$

ise:

$$\mathbb{E}[x^2] = \int_0^1 x^2 dx = \frac{1}{3}.$$

Monte Carlo'da çok sayıda (x_i) üretip karelerinin ortalamasını alıyoruz:

$$\frac{1}{N}\sum_i x_i^2 \longrightarrow = \frac{1}{3}$$

MC ile integral hesaplama: ∫x² dx

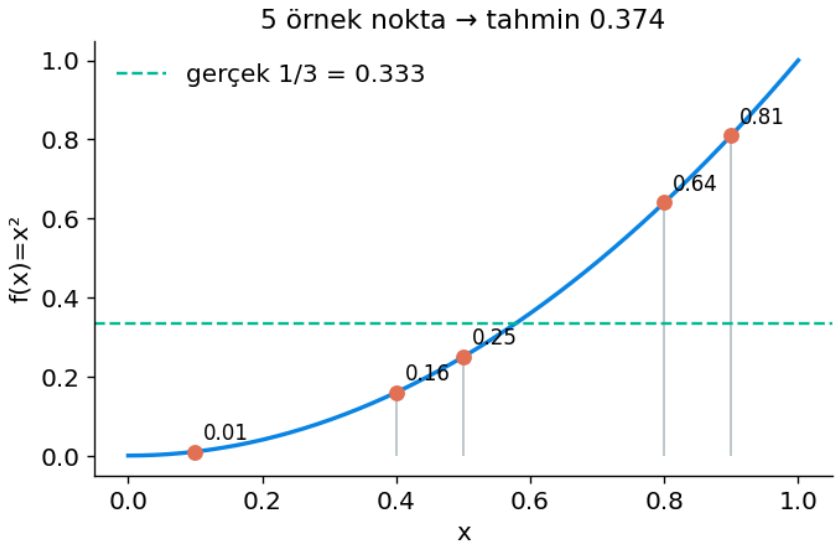
Az sayıda noktayla bile fikir çalışır; *N* arttıkça gerçek sonuca yakınsar.

Gerçek değer 1/3 ≈ 0.333.

x = 0.1, 0.4, 0.5, 0.8, 0.9
⟶ f(x) = 0.01, 0.16, 0.25, 0.64, 0.81.

5 örnekle tahmin: **0.374** — kaba ama doğru yönde.

$$\int_0^1 x^2\,dx = \frac{1}{3}, \qquad \hat{I}_5 = 0.374$$



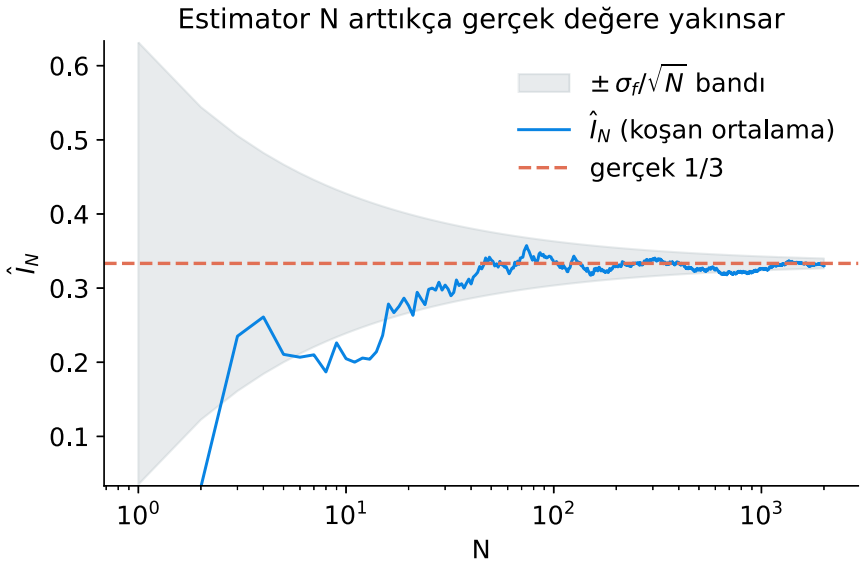
MC tahmin edicisi

$$\hat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Tahmin edici (estimator), $f(x_i)$ değerlerinin aritmetik ortalamasıdır (x_i'ler rastgele değişken).

N arttıkça tahmin gerçek integrale yakınsar.

Asıl güç **yüksek boyutlu**, **karmaşık tesir kesitli** integrallerde ortaya çıkar.



Örnek: π'nin değerini rastgele noktalar kullanarak bul

Bir kareye rastgele noktalar atıp çember içinde kalanları sayarak π tahmin edilebilir.

Şimdi aynı expectation-value fikrini, 0–1 değerli bir gösterge değişkeniyle uygulayalım.
Yani **hit-and-miss Monte Carlo**.

Birim kareye **uniform** nokta at; **çeyrek çember içinde kalanları** say.

$$\hat{\pi} = 4\,\frac{N_{\text{inside}}}{N_{\text{total}}}$$

Örnek: π'nin değerini rastgele noktalar kullanarak bul

Elimizde:

$$I = \int_0^1 f(x)\,dx$$

olsun ve $(0 \le f(x) \le 1)$.

Birinci yöntem: $x_i \sim U(0,1)$ üretip

$$I \approx \frac{1}{N} \sum_i f(x_i)$$

hesaplamak.

(**Ödev**: $\hat{\pi} \approx 4 \cdot \frac{1}{N} \sum_{i=1}^{N} \sqrt{1 - x_i^2}$ olduğunu gösteriniz)

Örnek: π'nin değerini rastgele noktalar kullanarak bul

İkinci yöntem: Birim kare içinde rastgele nokta üretiyoruz:

$$(x_i, y_i) \in [0,1]^2 .$$

Noktanın çeyrek çemberin **içinde** olma koşulu:

$$x_i^2 + y_i^2 < 1.$$

Bir gösterge değişkeni tanımlayalım:

$$Z_i = \mathbf{1}[x_i^2 + y_i^2 < 1] .$$

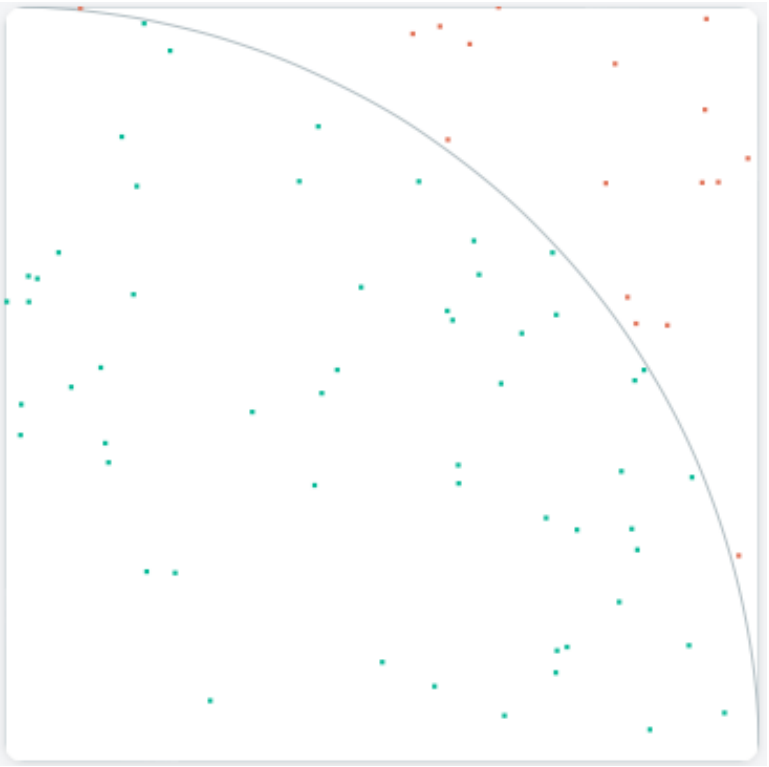
Bunun beklenen değeri, bir noktanın çeyrek çember içine düşme olasılığıdır:

$$\mathbb{E}[Z] = \frac{\text{çeyrek çember alanı}}{\text{kare alanı}} = \frac{\pi/4}{1} = \frac{\pi}{4} .$$

$$\frac{N_{\text{içeride}}}{N} \approx \frac{\pi}{4}$$

ve

$$\pi \approx 4\frac{N_{\text{içeride}}}{N} .$$



▶ **İnteraktif toy: d1_pi_estimation.html**

Soru: Hangi yöntem daha verimli?

Varyanstan Monte Carlo hatasına

Monte Carlo tahmin edicisi:

$$\hat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(X_i)\,.$$

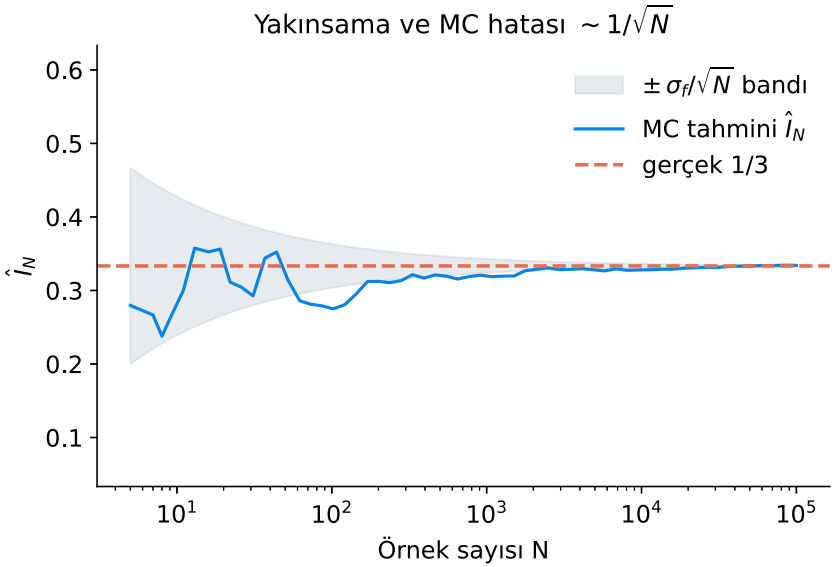
Her bir $f(X_i)$ rastgele değişkeninin varyansı σ_f^2 ise, ortalamanın varyansı:

$$\mathrm{Var}(\hat{I}_N) = \frac{\sigma_f^2}{N}\,.$$

Dolayısıyla MC standart hatası (örneklerden hesaplanan ortalamanın belirsizliği):

$$\delta_{\mathrm{MC}} = \sqrt{\frac{\mathrm{Var}(\hat{I}_N)}{N}} = \frac{\sigma_f}{\sqrt{N}} \propto \frac{1}{\sqrt{N}}$$

Not: $\mathrm{Var}[f(X)] = \mathbb{E}[f(X)^2] - \mathbb{E}[f(X)]^2\,.$



Ödev: $I = \int_0^1 x^2\,dx$ integralinin sayısal değerinin MC yöntemiyle tahmininde $N = 100$ ve $N = 10\,000$ nokta için MC hatasını hesaplayıp karşılaştırınız.

Rastgele ama tekrar üretilebilir

Aynı üretici kartı, yazılım sürümü ve tohum (seed), idealde aynı sözde-rastgele (pseudo-random) olay dizisini verir.

Aynı tohum → aynı dizi; hata ayıklama ve doğrulama için önemli.

Büyük üretimlerde tohum yönetimi önemlidir.

Aynı tohum ile paralel iş, yinelenen olay'lara ve örneklem korelasyonuna yol açabilir.

Aynı seed → aynı dizi (reproducibility)

```
rng = np.random.default_rng(seed=12345)
```

run 1:	0.741	0.028	0.396	0.512	0.885
run 2:	0.741	0.028	0.396	0.512	0.885

özdeş

Kümülatif Dağılım Fonksiyonu

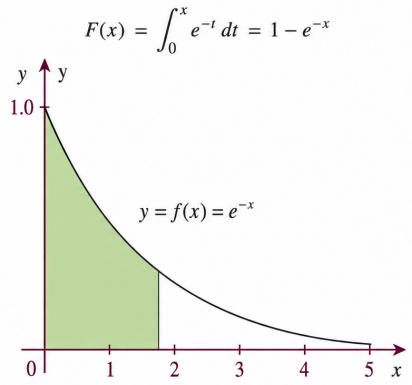
Bir rastgele değişkenin belirli bir değerden küçük veya eşit olma olasılığını verir:

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

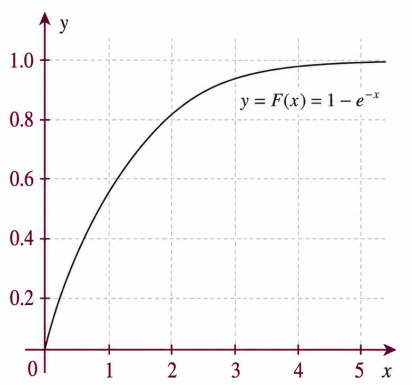
Sürekli bir dağılım için, olasılık yoğunluğu $p(x)$ ise:

$$F(x) = \int_{-\infty}^{x} p(t)\,dt\,.$$

Yani KDF, yoğunluğu soldan başlayarak biriktirir.



This is the graph of the exponential probability density function $f(x)$.



This is the graph of the exponential cumulative distribution function $F(x)$.

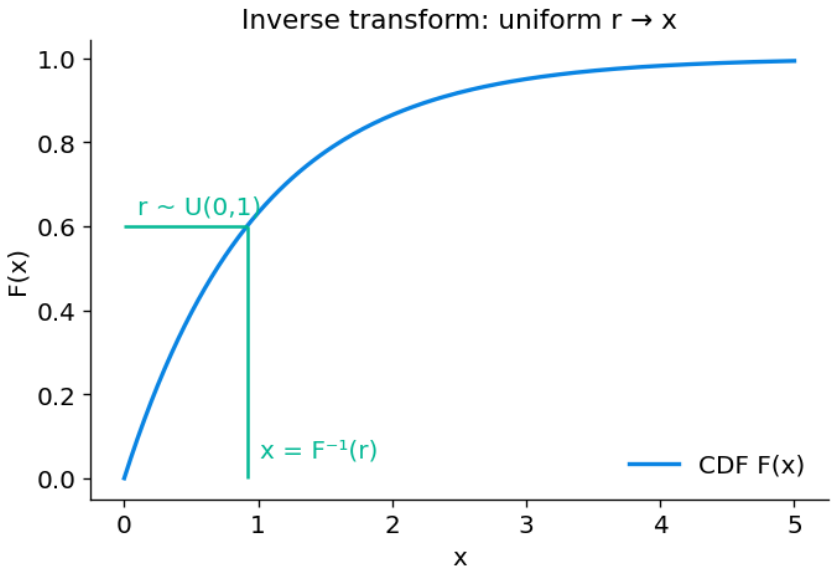
Dağılımlar: Yöntem 1: ters dönüşüm (inverse transform)

Kümülatif Dağılım Fonksiyonunun (KDF) tersi, düzgün (uniform) dağılımlı sayıyı istenen dağılıma taşır.

Düzgün dağılımla rastgele sayılar üretmek kolay; ama **üstel**, **Gaussian** veya **Breit–Wigner** dağılımı da isteriz.

F(x): olasılık yoğunluğunun **kümülatifi**. (KDF) $r \sim U(0,1)$ ise $x = F^{-1}(r)$, istenen $p(x)$'ten gelir.

$$F(x) = \int_{-\infty}^{x} p(t)\,dt, \qquad x = F^{-1}(r)$$



Ters dönüşüm örneği: üstel dağılım

Üstel dağılım, tek satırlık kapalı formülle örneklenir.

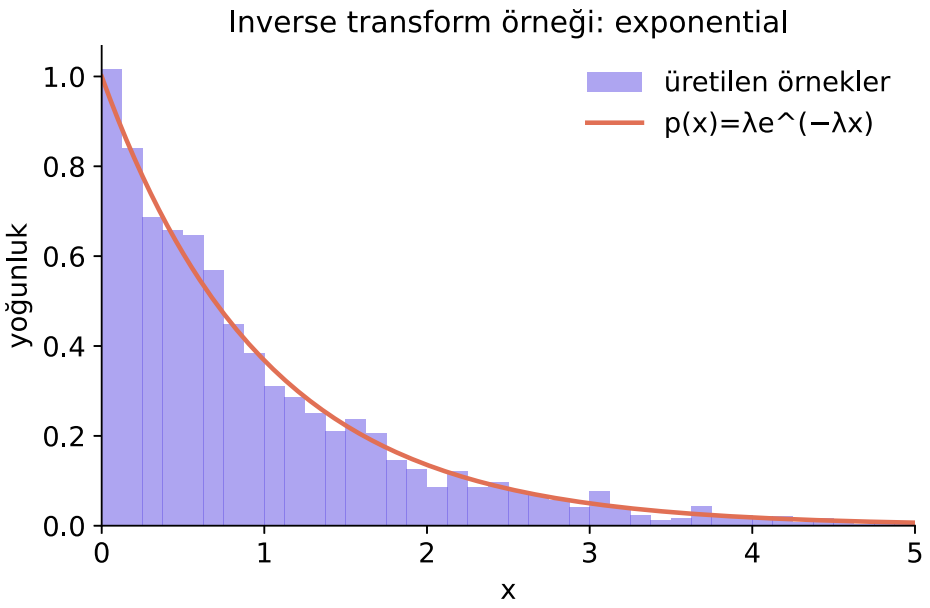
$p(x) = \lambda e^{-\lambda x}, x \geq 0;$

KDF: $F(x) = 1 - e^{-\lambda x}$.

Tersini al: $x = -(1/\lambda)\ln(1 - r)$.

Bozunma zamanı, serbest yol gibi büyüklükler böyle üretilir.

$$x = -\frac{1}{\lambda}\ln(1 - r)$$



Ödev: Kozmik müon geliş açıları

1. Deniz seviyesinde kozmik müon şiddetinin yaklaşık $I(\theta) \propto \cos^2\theta$ olduğunu kabul edin.
2. $d\Omega = \sin\theta\, d\theta\, d\phi$ kullanarak zenit açısının normalize edilmiş ODF'ını türetin.
3. ODF'den KDF'yi bulun.
4. F^{-1}'i hesaplayarak uniform rastgele sayılardan θ üretme formülünü bulun.
5. Python (veya diğer favori programlama dilinizi kullanarak) ile $N = 10^4$ kozmik müon üretin.
6. Üretilen θ değerlerinin histogramını teorik dağılımla karşılaştırın.
7. Azimut açısını ayrıca $\phi \sim U(0, 2\pi)$ üretin ve müonların geliş yönlerini çizdirin.

2. Yöntem: Kabul-ret

Eğrinin altına düşen rastgele noktaları kabul ederek istediğimiz dağılımı üretebiliriz.

KDF'nin tersi zor olduğunda işe yarar.

x üret, y üret; $y < f(x)$ ise kabul, değilse at.

Basit (ama verimsiz olabilir).

Ör: $f(x) = \frac{1}{2}\,\mathcal{N}(x; -2,1) + \frac{1}{2}\,\mathcal{N}(x; 2,1)\,.$

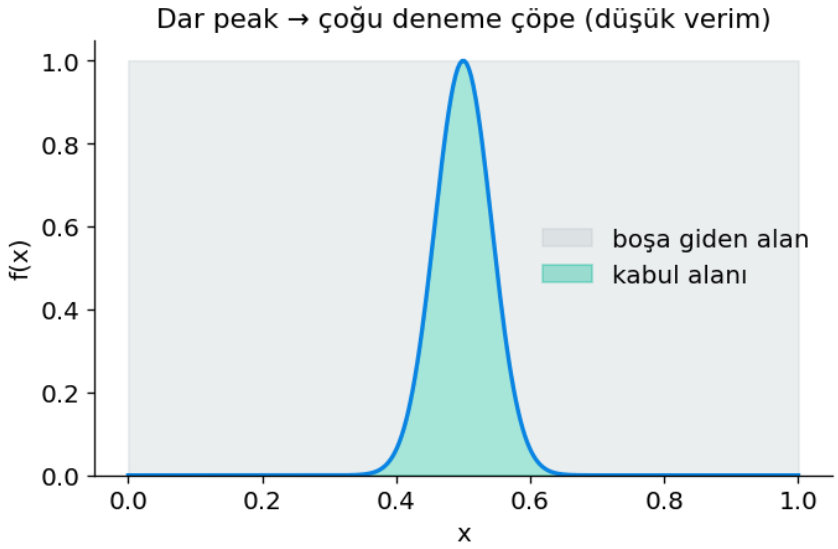
Accept–reject verimi

Fonksiyon sivri veya darsa verim düşer; çoğu deneme boşa gider.

Verim = kabul edilen / denenen.

Geniş boş alan = düşük verim
= harcanan CPU.

Çözüm: örnekleme dağılımını fonksiyona benzet → **importance sampling**.



$$\epsilon = \frac{N_{\text{accepted}}}{N_{\text{trial}}}$$

▶ İnteraktif toy: d1_accept_reject.html

Ödev — Kozmik müon üretimi II: Kabul–ret yöntemi

1. $p(\theta) = 3\cos^2\theta \sin\theta$ dağılımının maksimumunu ve maksimumun bulunduğu θ'yı bulun.

2. $\theta \sim U(0, \pi/2)$ ve $y \sim U(0, p_{\max})$ üreterek kabul–ret algoritmasıyla $N = 10^4$ kabul edilmiş müon üretin.

3. Üretilen θ değerlerinin histogramını kuramsal $p(\theta)$ ile karşılaştırın.

4. Algoritmanın verimini ölçün:

$$\epsilon = \frac{N_{\text{accepted}}}{N_{\text{trial}}}$$

5. Beklenen kuramsal verimi geometrik olarak hesaplayın ve simülasyonla karşılaştırın.

6. Sayfa 25'deki ters dönüşüm yöntemi sonuçlarıyla karşılaştırın: hangisi daha verimli ve neden?

Importance sampling: akıllı rastgelelik

Örnekleme dağılımını değiştirip ağırlıkla düzeltiriz; sonuç değişmez.

Her yere eşit bakmak zorunda değiliz.

Katkının büyük olduğu bölgeleri daha sık gez.

g(x)'ten örnekle; integrali f/g ile yeniden yaz.

$$I = \int \frac{f(x)}{g(x)}\, g(x)\, dx, \qquad x_i \sim g(x)$$

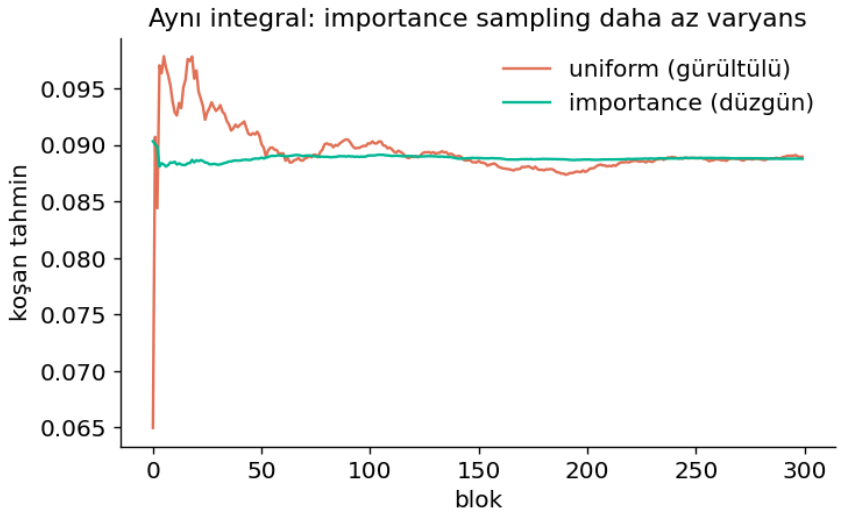
Importance sampling: tahmin edici ve ağırlık

Her örnek noktasına f/g ağırlığı verilir; bu düzeltme örnekleme yanlılığını önler.

Tahmin edici, f/g değerlerinin ortalamasıdır.
$w = f/g$: örnekleme dağılımı değişikliğinin düzeltmesidir.

Doğru yapılırsa **sonuç sabit**; sadece **varyans ve verim** değişir.

Geçerlilik koşulu: $f(x) \neq 0$ olan her yerde $g(x) > 0$ olmalı.



$$I = \mathbb{E}_g\left[\frac{f(X)}{g(X)}\right]$$

$$\hat{I}_N = \frac{1}{N}\sum_i \frac{f(x_i)}{g(x_i)} \qquad w_i = \frac{f(x_i)}{g(x_i)}$$

▶ İnteraktif toy: d1_importance_sampling.html

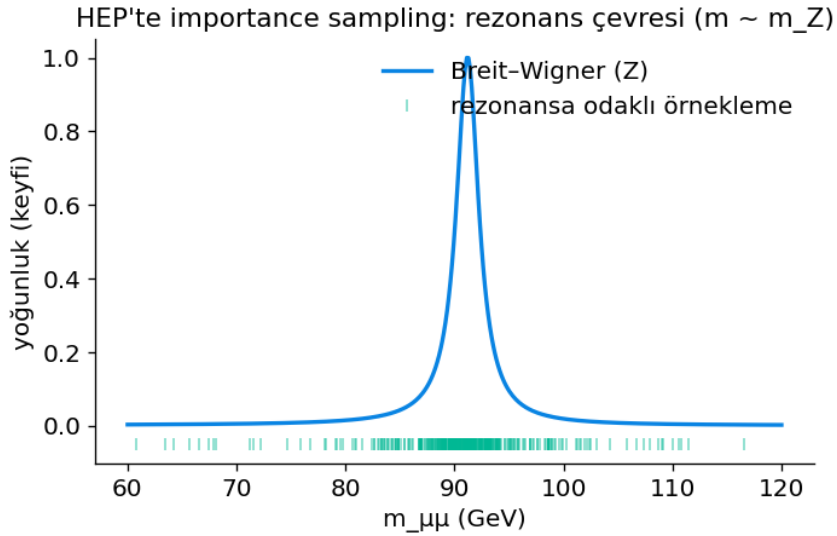
HEP'te importance sampling nerede?

Event generatorları, büyük katkılı veya analiz için nadir ve önemli bölgeleri hedefleyerek verim kazanır.

Rezonans çevresi: $m_{\ell\ell} \sim m_Z$

Soft/collinear bölgeler; ayrıca nadir ama önemli high-p_T tail.

Çok jetli final state'lerde farklı topolojiler; channel optimization.



[$g(x) \approx$ Breit-Wigner benzeri fonksiyon]

Nerede kaldık?

Zincirin yarısını kurduk: örnekleme ve akıllı örnekleme tamam; sıra ağırlıkta.

Tamam: integral → rastgele örnek → (inverse/accept–reject) → importance sampling.

Şimdi sırada **olay ağırlığı** kavramı var.

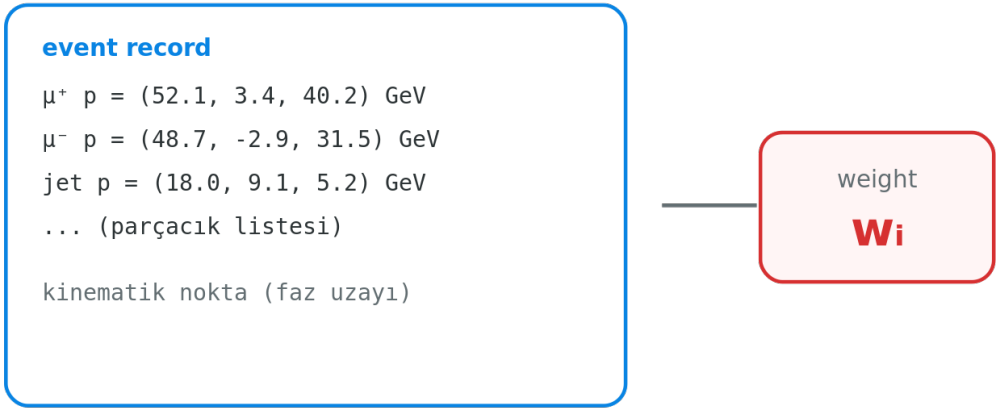
Olay ağırlığı (event weight) nedir?

Bir MC olay (event), bir kinematik nokta ile ona iliştirilmiş bir ağırlıktan oluşur.

olay$_i$ = (kinematik nokta$_i$, w_i).

Kinematik: momentumlar, parçacıklar, olay kaydı (event record).

Ağırlık: örnek noktanın tesir kesitine/yield'a katkısı.



Ağırlık normalizasyonu

Kesitte toplam mı ortalama mı kullanılacağı, olay ağırlığının normalizasyon tanımına bağlıdır.

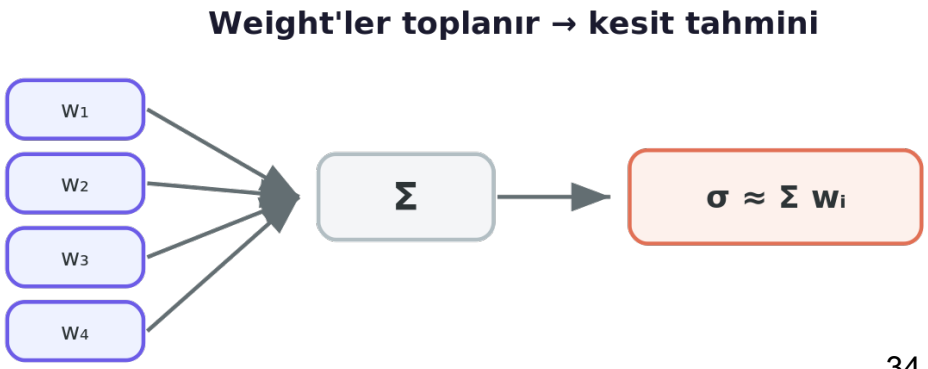
Importance weight $w_i^{imp} = f/g$ ise:

$$\hat{\sigma} = (1/N) \sum_i w_i^{imp}$$

1/N terimi w_i^{imp} içine absorbe edilirse olay ağırlığı w_i^{olay} elde edilir:

$$\hat{\sigma} = \sum_i w_i^{\text{olay}}$$

Histogram ve beklenen olay sayısı hesabında seçilen konvansiyon tutarlı kullanılmalıdır.



Ağırlık normalizasyonu

❗ Eğer 1/N faktörünü event weight'in içine dahil ediyorsak, weight **sample büyüklüğüne bağlı** hale gelir. Bu durumda

- **Örneklemeyi bölmek**: Parçaları ayrı ayrı işleyip histogramları sonunda toplamak sorun değil. Ama normalizasyonu güncellemeden parçaları tekil işleyip sonuçlarını kullanamayız.
- **Örneklemeye ekleme yapmak**: Sonradan örneklemeye olay eklemek N'yi değiştireceğinden ağırlıkların yeniden normalize edilmesini gerektirir.

Ağırlıklı histogram: bin içeriği

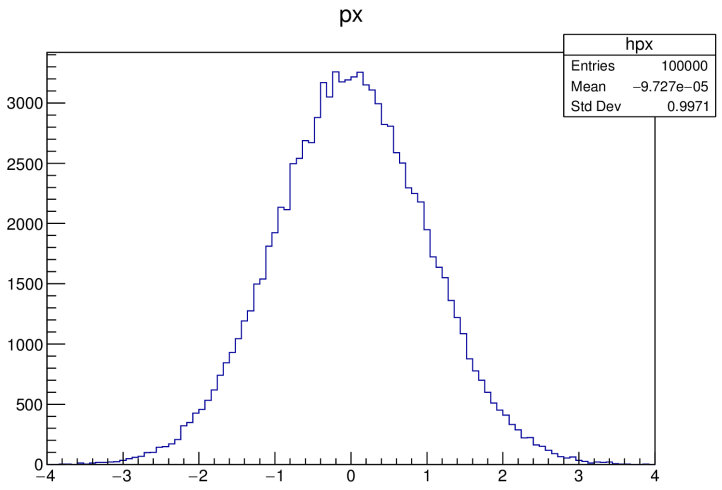
Histogramda olay saymayız; bin başına ağırlık toplarız.

Bu tartışmada w_i, global MC normalizasyonunu içeren saklanmış olay ağırlığıdır.

Bin içeriği: o bine düşen olayların ağırlık toplamı.

Örnek: bine giren weight'ler 2, 3, 5 → içerik 10.

$$H_b = \sum_{i \in b} w_i$$



Ağırlıklı histogram: bin hatası

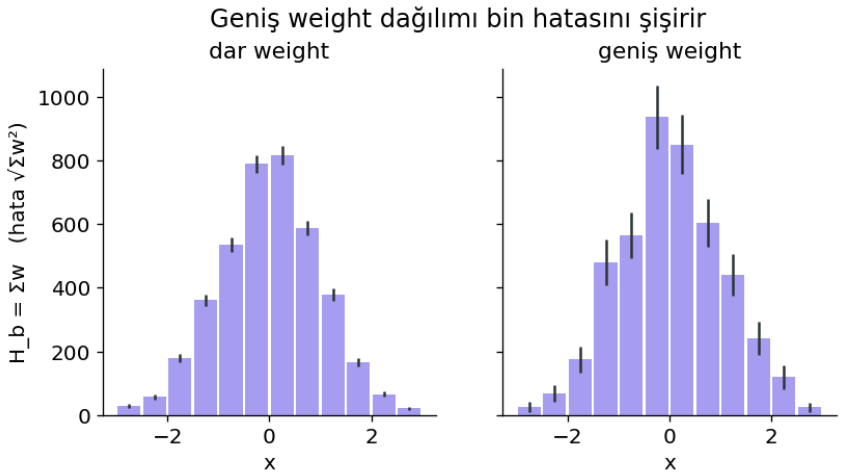
Ağırlıklı bir histogram bin'inin istatistik hatası, ağırlık karelerinin toplamının kare köküdür.

Bin hatası = $\sqrt{\sum w_i^2}$.

Tüm ağırlıklar w'ye eşitse hata $|w|\sqrt{N}$;
$w = 1$ için bildiğimiz $\sqrt{N}$ sayma hatasıdır.

* Birkaç büyük weight bin hatasını şişirir.

$$\delta H_b = \sqrt{\sum_{i \in b} w_i^2}$$



MC event ≠ gerçek olay

MC event sayısı ile beklenen fiziksel olay sayısı birbirinden farklıdır.

MC event sayısı: bilgisayarda üretilen örnek sayısı.

Beklenen olay sayısı: **ışınlık** (luminosite) ile normalize edilmiş fiziksel beklenti.

100 bin MC event, beklenen 100 bin gerçek olay demek değildir.

MC event ≠ gerçek olay

Örneğin gerçek CMS verisindeki bir çarpışma olayı, dedektörde gerçekten gerçekleşmiş bir proton-proton çarpışmasının kaydıdır.

MC olayı ise:

- seçtiğimiz kuramsal modelden,
- seçtiğimiz perturbatif mertebeden,
- PDF setinden,
- ölçek seçimlerinden,
- parton shower ve hadronizasyon modellerinden,
- dedektör simülasyonundan

üretilmiş **yapay bir örnektir**.

Dolayısıyla MC olayı “Doğada tam olarak bu olay gerçekleşti” olarak kabul etmek doğru değildir.

MC üretimi şunu söyler:

Kullandığımız model doğruysa, bu tür kinematik konfigürasyonlar belirtilen olasılık/kesit dağılımına göre ortaya çıkabilir.

! “Doğanın kopyası değil” ifadesi hakkında bir not: İyi bir olay üreticinin’ın amacı tam olarak doğada gözlenen **istatistiksel dağılımları** modellemektir. Tek bir olayı birebir kopyalamaz; **çok sayıda olayın oluşturduğu dağılımı** tahmin eder.

1 milyon olay her zaman 1 milyon olay değildir

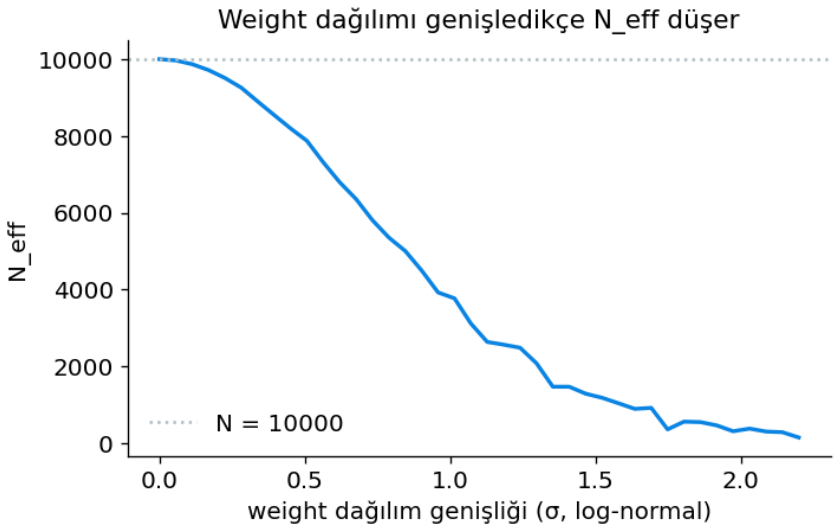
Ağırlıklı örneklemde asıl ölçü ham olay sayısı değil, efektif olay sayısıdır.

Kaç olay ürettiğin kadar, **fiziksel ağırlığın** bu olaylar arasında ne kadar **dengeli** dağıldığı da önemlidir.

N_{eff}, ağırlıkların dağılımını **tek sayıya** indirger.

Tüm ağırlıklar eşitse $N_{eff} = N$.

Ağırlıklar çok değişkense $N_{eff} \ll N$.



$$N_{\text{eff}} = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2}$$

N_{eff} : örnek

Aynı olay sayısı, çok farklı istatistiksel kalite verebilir.

Örneklem A: 1000 olay, hepsi $w = 1 \rightarrow N_{eff} = 1000.$

Örneklem B: 999 olay $w = 0.001$ + 1 olay $w = 100 \rightarrow N_{eff} \approx 1.02$

Olay sayısı aynı ama Örneklem B istatistiksel olarak **çok zayıf**.

$$N_{\text{eff}}^{A} = 1000, \qquad N_{\text{eff}}^{B} \approx 1$$

Olay üreticilerinde ağırlık çok temel ve yaygın bir kavramdır. Üretici hesaplaması çoğu zaman doğal olarak ağırlıklı olarak başlar; fakat pratik tam-simülasyon örneklemleri mümkünse unweight edilir. NLO ve sistematik varyasyonlarda ise olay ağırlıklarıyla karşılaşmak **son derece yaygındır**.

Unweighting: eşit ağırlıklı temsil

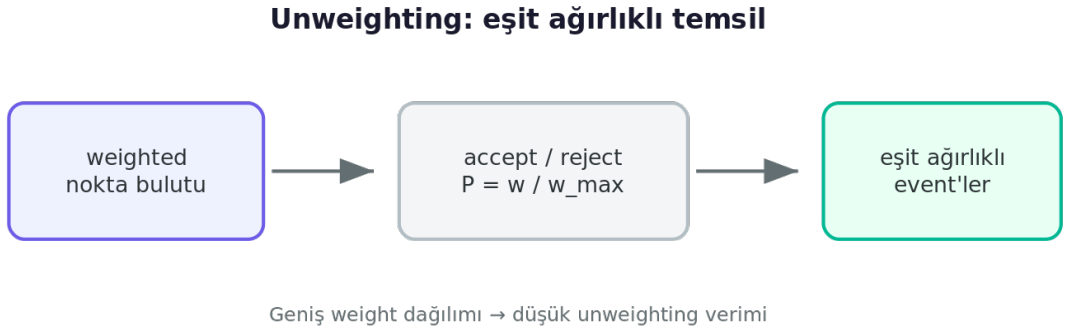
Ağırlığa orantılı kabulle eşit ağırlıklı bir event kümesi üretebiliriz.

Koşul: $0 \leq w_i \leq w_{max}$; kabul olasılığı $P = w_i / w_{max}$.

Kabul edilen olaylar eşit ağırlık taşır — analiz ve dedektör simülasyonu için pratik.

Ağırlıklı kuramsal olarak doğal; ağırlıksız zincir için kullanışlı.

$$P_{\text{accept}} = \frac{w_i}{w_{\max}}$$



Akılda kalacaklar

Monte Carlo yöntemi = rastgele örnekleme kullanarak sayısal bir niceliği tahmin etme veya bir olasılık dağılımından örnek üretme tekniği.

olay = ağırlıklı örnek nokta.

Histogramlarda olay saymayız, ağırlık toplarız.

Importance sampling sonucu değil, verimi ve varyansı değiştirir.

MC hatası 1/√N gider ama ağırlıklı örneklemde asıl mesele N_{eff}'tir.

Bu kavramlar faz-uzayı üretimi, unweighting, eşleştirme ve negative ağırlıkların temelidir.