H10: Monte Carlo

10.1 pseudo-random getal generatoren	
concept pseudo-random getal generator	= een manier om een rij getallen te produceren via een computer > is per definitie deterministisch, door hardware van computer > we willen echter een uniforme verdeling die zo random mogelijk is > probeer dit niet zelf te schrijven, gebruik gespecialiseerde bibliotheek (numpy, scipy) > tegenwoordig hebben generatoren een periode van 10^{43}
wenselijke eigenschappen PRNG	1: mogelijkheid om een seed te gebruiken dat de reeks random getallen start 2: mogelijkheid om laatste random getal in reeks op te slaan en te gebruiken als seed 3: reeks toevalsgetallen moet onafhankelijk zijn vh type computer / operating system >> PRNG genereert getallen r tussen 0 en 1
tests op PRNG	1: test van de uniformiteit: zijn alle waarden tss 0 en 1 even waarschijnlijk 2: test voor correlaties: bvb: komt na een klein getal altijd een groot getal 3: sequentietest: treden er reeksen op waarbij je van klein naar groot loopt? 4: gap-test: vellen er soms gaten in de uniforme distributie?
10.2 Monte Carlo en fysica	
10.2.1 simulatie van één detector	
vb: impuls van een deeltje	We kunnen de impuls ve deeltje bepalen door de kromming in een magneetveld te meten > kromtestraal wordt gemeten op verschillende plaatsen langsheen het spoor > onzekerheid vd impuls wordt bepaald door: - resolutie waarmee we de kromtestraal kunnen meten - de verstrooiing van de deeltjes in het materiaal van de detectoren De grootte van deze effecten wordt geschat via Monte Carlo > deeltjes met bepaalde impuls worden gevolgd doorheen detectiesysteem in magneetveld > - na doorgang van elke dunne laag materie wordt richting vh deeltje gewijzigd - gemeten positie in elke detector wordt lichtjes gewijzigd om resolutie te simuleren > complete set van gemeten coördinaten wordt gebruikt om impuls te schatten > bekijk dit getal tov de bekende startimpuls, dit is de totale resolutie >> hierdoor kunnen we in de ontwerpfase van de detector rekening houden hiermee > kunnen detector aanpassen zodat de resolutie zo groot mogelijk is
10.2.2 simulatie van één experiment	
Monte Carlo en experimenten	In een experiment wil je een bepaalde theorie bevestigen en andere ontkrachten > hoe weet je of een bepaald experiment dit wel degelijk kan aantonen? Simuleer groot aantal experimenten via Monte Carlo volgens beide theorieën > vergelijk de resultaten van de Monte Carlo experimenten van beide theorieën > als de resultaten overlappen, kunnen we de theorie niet verwerpen op basis van dit experiment > het heeft geen nut om het experiment uit te voeren
10.2.3 programma's testen	
programma testen	voor grote programma's is het lastig om elke lijn code te verifieren > test de code door opnieuw het experiment te simuleren met Monte Carlo > analyseer de gesimuleerde input met het programma

10.3 Methodes en algoritmes	
def: random getal u	In de technieken hieronder is u steeds een random getal uniform verdeeld tss 0 en 1
10.3.1 algemene methode: 'Hit and Miss' / verwerpingsmethode	
concept verwerpingsmethode	Stel: voor elke waarde van x is de wssheidsverdeling f(x) die we willen genereren gekend > sluit f(x) in in een vorm die C keer een eenvoudig te generen h(x) is ie: je moet het maximum van de functie $f_{max}(x)$ kennen merk op dat f(x) en h(x) genormeerd moeten zijn op 1 > dus C>1
def: verwerpingsmethode	1: genereer een kandidaat x op basis van h(x) 2: bereken f(x) en de hoogte van de 'enveloppe' C.h(x) 3: genereer een getal u en test of u.C.h(x) ≤ f(x) > als hieraan voldaan is wordt toevalsgetal x aanvaard, anders wordt ze verworpen en wordt de cyclus opnieuw gestart
10.3.2 inverse transformatie	
concept inverse transformatie	stel: je wilt toevalsgetallen genereren volgens een verdeling f(x) > zei $\mathcal{U}(0,1)$ een uniforme verdeling, dan: $$\begin{aligned} f(x)dx &= \mathcal{U}(0,1)dx \\ \int_{-\infty}^{x} f(t)dt = F(x) &= u \end{aligned}$$ met F(x) de cumulatieve verdeling
st: inverse transformatie	De vergelijking F(x) = u kan opgelost worden naar x om te vinden: $$x = F^{-1}(u)$$ > je kan een toevalsgetal u_j genereren volgens een uniforme verdeling > gebruik $F^{-1}(u_j)$ om de corresponderende x_j te vinden > deze variabele is verdeeld volgens f(x)
inverse transformatie voor discrete functie	Als f(x) discreet is, vertoont F(x) discontinue sprongen van grootte $f(x_k)$ bij elke toegelaten waarde van x > kies dan u volgens een uniforme verdeling en vindt x_k zodanig dat: $$F(x_{k-1}) < u \leq F(x_k) = \sum_{i=1}^{k} f(x_i)$$
10.3.3 toepassing op belangrijke verdelingen	
sin en cos ve toevalshoek	genereer een sinus via: $F(\phi) = \int_0^{\phi} \sin\theta d\theta = -\cos\phi = u$ dus: $\phi = \arccos(-u)$. > dit is best traag op een computer sneller: verwerpingsmethode • Genereer hiervoor twee getallen u_1 en u_2 , uniform tussen 0 en 1. • De nieuwe variabele $v_1 = 2u_1 - 1$ is uniform verdeeld tussen -1 en +1. We hernoemen $v_2 = u_2$ tussen 0 en 1. • We berekenen $r^2 = v_1^2 + v_2^2$. Indien $r^2 > 1$ begin je opnieuw. > genereer punten die uniform verdeeld zijn op opp. van cirkel met straal 1 > je zou denken dat de sin en cos vd poolhoek ϕ volgt uit: $$v_1 = r\cos\phi \qquad v_2 = r\sin\phi$$! maar v_2 is positief en zal dus enkel hoeken tss 0 en π opleveren > om dit te verhelpen moeten we de hoek verdubbelen: $$S = 2\frac{v_1 v_2}{r^2} \qquad C = \frac{v_1^2 - v_2^2}{r^2}$$

Gauss verdeling	We kunnen de inverse transformatie niet gebruiken nl: cumulatieve verdeling ve Gauss is niet analytisch op te schrijven bekijk: $\mathcal{N}(z) = \frac{1}{2\pi} e^{-z^2/2} \qquad (-\infty \leq z \leq +\infty)$ een variabele die x verdeeld is volgens een Gauss met gemiddelde μ en standaardafw. σ > is hier eenvoudig af te leiden via de transformatie x = μ + zσ Neem 12 uniform verdeelde toevalsgetallen > tel ze op en trek er 6 van af > centraal limiet theorema garandeert dat dit een Gauss verdeling wordt > deze methode is wel beperkt tot 6σ in de staarten, waar er slechte bendering is >> we kunnen beter: zie p229
Poisson verdeling	• Begin met $k = 1$ en stel $A = 1$. • Genereer u. • Vervang A door uA. ▷ Als nu $A < e^{-\lambda}$ (met λ de Poisson gemiddelde waarde) dan accepteer je $n_k = k - 1$ en stop. ▷ In het andere geval verhoog je k met 1, genereer een nieuwe u en begin opnieuw, steeds met de waarde van A uit de vorige poging.
10.3.4 multi-dimentionele verdelingen	
punten op een schijf	We genereren uniform verspreide punten in een cirkel met straal R > met dichtheid ρ punten per oppervlakte-eenheid Werk in poolcoördinaten 0≤r≤R en 0≤ϕ≤2π > de wssheidsdichtheden van r en ϕ zijn het aantal punten dat in de infinitesimale oppervlakken ligt, vergeleken met het totaal aantal in de cirkel: $p(\phi)d\phi = \frac{\rho R^2 d\phi/2}{\rho\pi R^2} = \frac{d\phi}{2\pi} \qquad q(r)dr = \frac{\rho\, 2\pi\, r dr}{\rho\pi R^2} = \frac{2r}{R^2}dr$ De corresponderende cumulatieve distributies zijn dan: $u_1 = P(\phi) = \int_0^\phi p(\phi')d\phi' = \frac{\phi}{2\pi} \qquad u_2 = Q(r) = \int_0^r q(r')dr' = \frac{r^2}{R^2}$ De inversie geeft dan: $\phi = 2\pi u_1 \qquad r = R\sqrt{u_2}$ waarbij u_1 en u_2 twee (onafhankelijke) toevalsgetallen zijn die uniform verdeeld zijn tussen 0 en 1.
punten op boloppervlak	we genereren uniform verdeelde punten op het opp. ve bol met straal R > zet om in bolcoord: $x = R\sin\theta\cos\phi \qquad y = R\sin\theta\sin\phi \qquad z = R\cos\theta$ > alle punten liggen op afstand R, dus dit is geen variabele, enkel ϕ en θ > beschouw een infinitesimaal klein opp. vd bol met als oppervlakte $d\Omega = \sin\theta d\theta d\phi$ zei n het totaal aantal punten en dn het aantal punten in dΩ, dan: $\frac{n}{4\pi} = \frac{dn}{d\Omega}$ > de wssheid dat een punt in dΩ ligt is dus: $p(\Omega)d\Omega = \frac{dn}{n} = \frac{d\Omega}{4\pi} = \frac{\sin\theta d\theta d\phi}{4\pi}$ de wssheidsdichtheden voor θ en ϕ vinden we door te integreren over de andere hoek $p(\phi)d\phi = \frac{1}{4\pi}\int_0^\pi \sin\theta d\theta = \frac{1}{2\pi}d\phi \qquad q(\theta)d\theta = \frac{1}{4\pi}\sin\theta d\theta \int_0^{2\pi} d\phi = \frac{\sin\theta}{2}d\theta$ De cumulatieve verdelingen worden dan: $u_1 = P(\phi) = \frac{\phi}{2\pi} \qquad u_2 = Q(\theta) = \frac{1-\cos\theta}{2}$ en de formules voor het genereren van de toevalshoeken ϕ en θ worden: $\phi = 2\pi\, u_1 \qquad \theta = \arccos(1 - 2u_2)$ waarbij u_1 en u_2 twee (onafhankelijke) toevalsgetallen zijn die uniform verdeeld zijn tussen 0 en 1.

10.4 Monte Carlo en statistische methodes

10.4.1 gecorreleerde variabelen	
gecorreleerde variabelen in Monte Carlo	Soms hebben we gecorreleerde punten nodig vb: simulatie ve bundel deeltjes die vanuit benaderde focus vertrekt > hoek θ van deeltje zal gecorreleerd zijn aan transversale positie y: $\begin{cases} \theta = x_1 \cos\alpha + x_2 \sin\alpha \\ y = -x_1 \sin\alpha + x_2 \cos\alpha \end{cases}$ de covariantie wordt dan gegeven door: $\mathrm{cov}(\theta, y) = \sin\alpha\cos\alpha(\sigma^2(x_2) - \sigma^2(x_1))$ (10.16) Er zullen dus correlaties optreden als de spreiding op x_1 en x_2 verschillend is en als zowel $\cos\alpha$ als $\sin\alpha$ verschillend zijn van 0.
10.4.2 complexe verdelingen en foutenpropagatie	
foutenpropagatie via Monte Carlo	In H4 zagen we dat de fout op een variabele gepropageerd wordt tot een fout in de functie > daar hebben we gesteld dat de fties weinig wariëren in het gebied van belang Deze veronderstelling is niet altijd juist > bovendien: we zijn vaak ook geïntereseerd in de vorm vd verdeling vd functiewaarde > en dus niet in de standaardafwijking ervan > bvb: belangrijk om de significantie ve bepaald resultaat te bepalen > MC geeft een manier om verder te komen:
vb: MC in foutenpropagatie	veronderstel een verdeling vd variabele Z die we willen kennen, gegeven door: $Z = \frac{\sin X}{\sin Y}$ met x en y twee Gaussisch verdeelde hoeken met gemm. en standaardafw.: $\theta_x = 20^\circ \qquad \sigma_x = 3^\circ$ $\theta_x = 13^\circ \qquad \sigma_x = 3^\circ$ de formules voor foutenpropagatie geven de volgende info over de onbekende verdeling van Z: $< Z >= \frac{\sin\theta_x}{\sin\theta_y} = 1.52$ $\sigma_z = \left[\left(\frac{\cos\theta_x}{\sin\theta_y}\right)^2 \left(\frac{\pi}{180}\right)^2 \sigma_x^2 + \left(\frac{\sin\theta_x \cos\theta_y}{\sin^2\theta_y}\right)^2 \left(\frac{\pi}{180}\right)^2 \sigma_y^2\right]^{\frac{1}{2}} = 0.41$ Echter, gebruik nu Monte Carlo > genereer een groot aantal x en y waarden volgens de Gauss verdeling en bereken Z > we vinden het gemm. en standaardafw.: $\mu_z = 1.61 \qquad s_z = 0.49$ wat afwijkend is van de vorige resultaten Bovendien vinden we dat 75% van de punten tss μ_z-s_z en μ_z+s_z ligt > is veel meer dan voor een Gauss verdeling > de MC benadering is beter dan de Gauss benadering via formules van foutenpropagatie
10.4.3 confidentie-intervallen	
berekenen van confidentie-interval via MC	Om confidentie-interval te bepalen is er kennis van de verdeling nodig > we kunnen nu via MC eender welke verdeling bepalen (nl door de vorige paragraaf) > in werkelijkheid is dit heel omslachtig werk, in plaats daarvan: gebruik Bootstrap methode
Bootstrap methode	bepaal de verdeling van z en bepaal de grenzen waarvoor er 68,3% kans is dat z erbinnen ligt > op die manier gebruikt men enkel info die beschikbaar is uit het experiment > we vermijden onduidelijkheid over de keuze van x en y > gevolg: resultaat wijkt af van het correcte confidentie-interval

10.5 numerieke integratie	
10.5.1 Monte Carlo integratie	
niet-MC integratie	klassiek zouden we numeriek een integraal oplossen door het interval tss a en b op te delen in n gelijke delen en als benadering voor I de som: $$I = \int_a^b y(x)dx \quad \longrightarrow \quad I \approx \frac{b-a}{n}\sum_{i=1}^{n} y(x_i)$$ hierbij is $x_i = a + (i + \frac{1}{2})\frac{(b-a)}{n}$
def: MC integratie	De Monte Carlo manier evalueert dezelfde som > ipv $y(x_i)$ bij gelijk gespaciëerde x-waarden te berekenen, nemen we nu toevallige x_i: $$x_i = a + r_i(b-a)$$ waarbij r_i **toevalsgetallen zijn die uniform verdeeld zijn tussen 0 en 1**.
10.5.2 nauwkeurigheid van Monte Carlo berekeningen	
st: nauwkeurigheid van MC	als we n toevalsgetallen nemen om de integraal I te schatten > dan is de nauwkeurigheid van de methode evenredig met 1/√n > we vinden namelijk: de schatting van I via MC heeft nauwkeurigheid σ/√n
voordeel MC methode	- voor 1D integralen is MC-integratie zeer onnauwkeurig vergeleken met andere methoden > echter: voor meer-dimensionale integralen wordt de nauwkeurigheid van andere methoden steeds slordiger > voor MC-integratie blijft die 1/√n - vaak zijn de grenzen van meer-dimensionale integralen zeer complex > lastig om te integreren via klassieke methoden > voor MC moeten we enkel checken of gegenereerde punt binnen de grenzen ligt > heel makkelijk
> hoe groter n, hoe langer de MC methode duurt om te computeren via de pc > de nauwkeurigheid stijgt trager en trager naarmate n groter > we moeten MC efficiënter maken	
stratificatie	verdeel het integratiegebied in 2 gelijke gebieden in x > genereer de helft van de toevalspunten in elk vd twee deelgebieden > je krijgt een meer uniforme verdeling van x-waarden > variantie verkleint
niet-uniforme sampling	We willen x-punten met hogere densiteit waar het integrand het snelst verandert > verdeel x-gebied in twee ongelijke gebieden > kies het aantal punten in elk gebied zodat de variantie minimaal wordt
importance sampling	variantie van MC uitkomst is evenredig met die van het integrand > transformeer de integraal zodat deze een kleinere spreiding heeft: $$\int y(x)dx = \int \frac{y(x)}{w(x)}w(x)dx = \int \frac{y(x)}{w(x)}dv(x)$$ waarbij $$v(x) = \int w(x)dx$$ variantie vh antwoord hangt nu af van die van y(x)/w(x) in plaats van y zelf > kies de vorm van w dicht bij y, zodat de variantie klein wordt
antithetische variabelen	voor elk MC gegenereerd punt x nemen we een ander punt x' > gebruik de bijdrage y(x)+y(x') tot de integraal > als we x' goed kiezen kunnen y(x) en y(x') anticorrelatie vertonen > variantie op y(x)+y(x') wordt verkleind
>vb van alle methoden: zie p237	