Project 3 - Computational physics - FYS3150

Department of Physics, University of Oslo, Norway.
Philip Niane og Rohullah Akbari
Link til githubmappen:
https://github.com/philipkarim/Philip-and-Rohullah-ComFys

Abstrakt

I denne rapporten er det vist hvordan energien mellom to elektroner i et helium atom i grunntilstanden kan utregnes ved forskjellige integrasjonsmetoder. Integrasjonsmetodene som er brukt er: Gauss Legendre kvadratur som gir en relativ stor feil, med mindre integrasjonsstegene N er mange nok. Gauss Laguerre kvadratur som gir bedre resultater for små N enn Legendre, men fra erfaring da kodene ble kjørt er begge metodene relativt tidskrevende ved store N. "Brute force" Monte Carlo er også brukt men her ble presisjonen relativt dårlig sammenliknet med en forbedret Monte Carlo. Den forbedrede Monte Carlo metoden, også kalt importance sampling gir gode approksimasjoner men kan være rimelig tidskrevende. Denne metoden ble modifisert ved bruk av parallellisering noe som førte til en veldig tidseffektiv metode.

Innholdsfortegnelse

1 Introduksjon

Målet med dette prosjektet er å løse er å integrere ved bruk av forskjellige numeriske metoder. De numeriske metodene skal naturligvis programmeres og sammenliknes. Integralet som skal løses er integralet som gir grunntilstandsenergien mellom to elektroner i et helium atom:

$$\langle\frac{1}{|r_1 - r_2|}\rangle = \int_{-\infty}^{\infty} dr_1 dr_2 e^{-2\alpha(r_1, r_2)} \frac{1}{|r_1 - r_2|} \tag{1}$$

Hvor

$$\mathbf{r}_i = x_i \mathbf{e}_x + y_i \mathbf{e}_y + z_i \mathbf{e}_z \tag{2}$$

Som gir

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}. \tag{3}$$

Konstanten α blir satt til å være lik 2, da dette tilsvarer ladningen til et helium atom, Z=2.
Metodene som skal ses på i dette prosjektet er Gauss Legendre, Gauss Laguerre og Flere versjoner av Monte Carlo integrasjon. Det blir brukt både kartesiske koordinater og kule koordinater. Prosjektet gir også trening i bruk av paralellisering av kode som gir programmer som kjøres enda raskere.[3]

2 Teori

2.1 Gauss Kvadratur

En av integrasjonsmetodene som skal brukes er Gauss kvadratur. Denne metoden skiller seg ut fra en del andre metoder innenfor numeriske metoder ved at steglengden mellom x_i og x_{i+1} ikke er fast. Gauss kvadratur går ut på å skrive om integranden til følgende form:

$$A = \int_a^b f(x)dx = \int_a^b W(x)g(x)dx \approx \sum_{i=0}^{N-1} \omega_i g(x_i) \tag{4}$$

Hvor W(x) er vektfunksjonen, samtidig som aproksimasjonen blir nøyaktig om graden til polynomfunksjonen g(x) er mindre enn 2N-1. Vektene w_i fås gjennom de ortogonale polynoms-algoritmene som for eksempel Legendre, Laguerre, Hermite og Chebyshev. I dette prosjektet fokuseres det kun på Legendre og Laguerre. [1]

2.1.1 Gauss Legendre Kvadratur

Gauss Legendre metode bruker finite integrasjonsgrenser. Vektfunksjonen til Legendres metode W(x)=1, derfor er det mulig ut ifra likning (4) å se at metoden kan kjøres på integralet uten å måtte skrives om. Da x_i, y_i og z_i defineres fra $-\infty$ til ∞ vil passende finite grenser settes senere i rapporten.

2.1.2 Gauss Laguerre Kvadratur

Gauss Laguerre metode er fin å bruke når det opereres med grenser som går mot uendelig. Dette passer bra med tanke på at en endring til sfæriske koordinater vil gi r $\in$ [0,∞). Samtidig blir vinkelgrensene $\theta \in$[0,π] og $\phi \in$[0,2π].
Laguerre har en vektet funksjon på følgende form: $W(x) = x^\alpha e^{-x}$. Dette betyr at integralet som skal utregnes må skrives om på følgende form:

$$A = \int_0^\infty x^\alpha e^{-x} g(x)\, dx = \sum_{i=1}^{N} \omega_i g(x_i).$$

Etter uttrykket er skrevet om til sfæriske koordinater og skrevet om til formen over, kan uttrykket skrives på følgende måte: (Utledning kan ses i appendix)

$$A = \frac{1}{(2\alpha)^5} \int_0^\infty \int_0^\infty \int_0^\pi \int_0^\pi \int_0^{2\pi} \int_0^{2\pi} W(u_1) W(u_2) g(u, \theta, \phi)\, du_1 du_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2$$

Hvor

$$W(u_i) = (u_i)^2 e^{-u_i} \tag{5}$$

og

$$g(u_1, u_2, \theta_1, \theta_2, \phi_1, \phi_2) = \frac{\sin\theta_1 \sin\theta_2}{r_{12}} \tag{6}$$

r_{12} og $cos(\beta)$ er oppgitt i oppgaven som:

$$\frac{1}{r_{12}} = \frac{1}{\sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\beta)}}, \tag{7}$$

og

$$\cos(\beta) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2)\cos(\phi_1 - \phi_2)). \tag{8}$$

Dermed er uttrykket klart, og radiusdelen kan kjøres gjennom en Laguerre funksjon da radiusen går mot ∞, mens det holder å kjøre vinkeldelene gjennom en Legendre funksjon. Da disse grensene er begrenset.

2.2 Monte Carlo

Monte Carlo er den andre integrasjonsmetoden som skal brukes. Denne metoden er basert på middelverdisetningen fra Kalkulus, hvor middelpunktet til en funksjon skal finnes mellom punktene a og b. Ved å utføre denne metoden uendelig mange ganger så vil resultatet konvergere mot middelveriden av funksjonen.

Men siden det lar ikke å utføre denne metoden uendelig mange gang er så fører det til en feil. Variansen, med hensyn på N, er gitt ved:(Utledning kan ses i appendiks)

$$\sigma_N^2 = \frac{1}{N}(\langle f^2 \rangle - \langle f^2 \rangle)$$

Standardavvik, eller *standard deviation*, er proposjonal med den inverse kvadratroten av N antall prøver, $\sigma_N = \frac{1}{\sqrt{N}}$.

2.2.1 Brute-kraft og uniform distribusjon

I denne delen blir funksjonen integrert med en brute-kraft. Med dette menes det en uniform distribusjons funksjon. Det blir ikke tatt noen spesielle hensyn til for å løse integralet, men isteden blir det lagd tilfeldige tall mellom grensene som brukes for å løse integralet. Vi kan se på integralet på følgendevis:

$$A = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(x1, y1, z1, x2, y2, z2) dx_1 dy_1 dz_1 dx_2 dy_2 dz_2$$

Grensene til dette integralet går fra uendelig til uendelig. Det gjør at det blir umulig å løse dette. For å kunne gjøre dette integralet enklere så begrenser vi grensene til et bestemt tall istedenfor uendelig, altså $[-\infty, \infty] \to [a, b]$. I kan da tilnærmes til: (Se appendiks for utledning)

$$A \approx (b-a)^6 \frac{1}{N} \sum_{i=1}^{N} f(x_{1i}, y_{1i}, z_{1i}, x_{2i}, y_{2i}, z_{2i})$$

der $(b-a)^6$ er Jacobi-determinanten og alle $x_{1i}...z_{2i}$ er uniforme distributerte tilfeldige tall. Disse tallene er produsert ved likning:

$$s_i = a + (b-a)x_i \tag{9}$$

Der $s_i \in [x_{1i}, ..., z_{2i}]$ og x_i er et helt tilfeldig tall mellom 0 og 1.

2.2.2 Forbedret Monte Carlo og Importance sampling

For å kunne å forbedre metoden ovenfor så er det mulig å velge en annen distribusjonsfunksjon. I dette tilfeldig velges det en funksjon som følger integralet nøye. Siden integralet inneholder en eksponentiall del så er det lurt å velge en eksponentiell distribusjonsfunksjon. Vi transformerer fra kartesiske til sfæriske koordinater. Og da velger følgende eksponentiellfunskjon $p(r) = le^{-rl}$ der l er en konstant. Her har det blitt brukt $\cos(\theta)_i$ som variabel og ikke θ_i. Da får vi følgende variabler:

$$r_i = -\frac{1}{l}\ln(1 - x_i),$$

$$\cos(\theta)_i = 2x_i - 1$$

$$\phi_i = 2\pi x_i$$

der $r_i \in [0,\infty)$, $\cos(\theta)_i \in [-1,1]$, $\phi_i \in 2\pi[0,-1]$ og x_i er et tilfeldig tall mellom 0 og 1. Funksjonen blir da:(Se appendiks for utledning)

$$f(x_{1i}, y_{1i}, z_{1i}, x_{2i}, y_{2i}, z_{2i}) = f(r_{1i}, r_{2i}, \cos(\theta)_{1i}, \cos(\theta)_{2i}, \phi_{1i}, \phi_{2i}) = \frac{r_{11}r_{22}}{r_{12}}$$

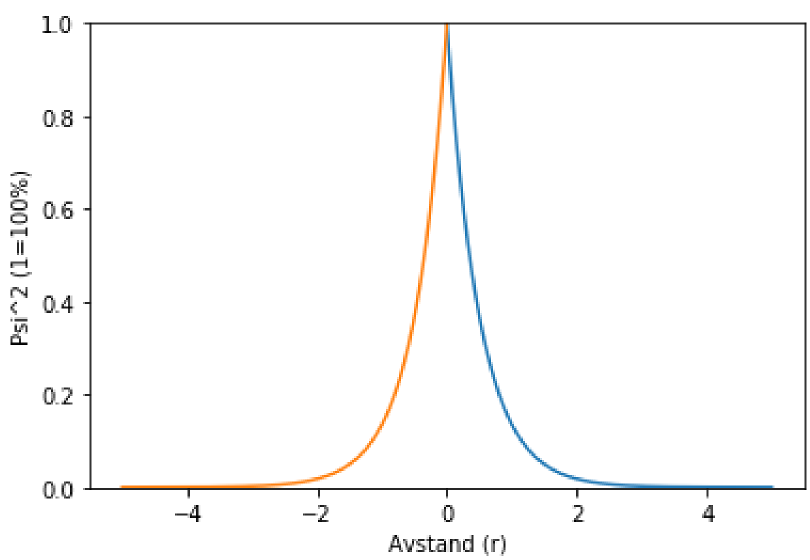
Dette lar seg kun gjøre hvis l = 4. Og integralet kan da tilnærmes til:

$$A \approx (2\pi)^2 2^2 \frac{1}{A^2}\frac{1}{N}\sum_{i=1}^{N}\frac{r_{11i}r_{22i}}{r_{12i}} \tag{10}$$

3 Metode

3.1 Gauss Kvadratur

Som sagt er det litt vanskelig å beregne integraler for uendelige grenser ved bruk av Legendres metode. Dermed må det bestemmes definerte grenser å bruke. Grensene bestemmes ved å plotte bølgefunksjonen $\Psi(r_1 r_2) = e^{-\alpha(r_1+r_2)}$. Plottet av bølgefunksjonen er fremstilt i figur 1.



Figur 1: Sannsynlighetsdistribusjon for bølgefunksjonen $\Psi(r_1 r_2) = e^{-\alpha(r_1+r_2)}$. Grensene går mot 0, ved større avstand mellom elektronen.

Utifra figur 1 konvergerer grafen mot 0, grafen når $\approx$ når 0 når r ligger utenfor intervallet [-3,3]. Dette blir integrasjonsgrensene i Legendres metode.

3.1.1 Gauss Kvadratur- Kartesiske Koordinater

Koden som brukes til Legendre integrasjon er inspirert av eksempelkodene fra github[2].
Ved bruk av kartesiske koordinater holder det å bruke Legendres metode, med integrasjonsgrenser fra -3 til 3. Når det gjelder psudokoden kjøres Legendre funksjonen med de oppgitte grensene for å lage Legendrepolynomet. Videre er integranden lik det som ble oppgitt i oppgaveteksten. Denne integranden blir satt opp ved hjelp av en funksjon. Om avstanden mellom elektronene nærmer seg 0, vil funksjonen som lager uttrykket returnere 0 ved hjel av en if/else setning. Videre vil dette uttrykket brukes i en seksdimensional for-loop for å regne ut vektene. Vektene blir videre brukt i samme loop ved å multiplisere disse med integranden som ble laget i starten.

3.1.2 Gauss Laguerre- Sfæriske Koordinater

Koden som brukes til Laguerre integrasjon er stort sett inspirert av eksempelkodene fra github[2].
Ved bruk av sfæriske koordinater er det som sagt tidligere i rapporten kun nødvendig å bruke laguerre på radialdelen, mens Legendre brukes på vinkelene på samme måte som med de kartesiske koordinatene, men med annerledes grenser naturligvis.
Psudokoden har store likheter som over, på samme måte, blir integranden produsert ved bruk av en funksjon. Også her returnerer funksjonen 0 om avstanden mellom elektronene blir for liten, ved hjelp av en if/else setning. Laguerrepolynomet produseres ved å kjøre radialdelen gjennom Laguerre funksjonen. Videre brukes en seksdimensional loop i dette tilfellet også. Den seksdimensionale loopen fungerer på samme måte som for Legendre over, bortsett fra at denne brukes på sfæriske koordinater. Koden kjøres og bruker legendre på vinklene og Laguerre på radialdelen som til slutt returnerer en approksimasjon til integralet.

3.2 Monte Carlo metoden

Koden og implementasjon av denne metoden er stor inspirert av eksemplene i Lecture Notes. Programmet starter med å definere de forskjellige variablene. Deretter blir det definert en vektor y som blir brukt til å lagre de tilfeldige tallene. Den aller viktigeste koden i dette programmet er nemlig løkken. Denne løkken starter med å finne tilfeldige tall mellom 0 og 1, ved bruk av funksjon *randu*, og disse blir lagret i x vektoren. Deretter blir x-vektoren brukt til å finne de andre variablene som inngår integranden. Så blir de variablene brukt til å rope integraden og dermed lagret i fx. fx blir summert opp i $crude_{mc}$ og kvadrat av fx blir summert i $sum - sigma$. Deretter berenges integralet og variansen ved bruk av definisjonene ovenfor i teoridel.
Et punkt som er viktig å legge merke til er at vi fikk ulike verdier for integralet med samme N hver gang vi kjørte programmet. Dette førte også til at vi fikk ulike verdier for relativ feil, variansen og tiden. Vi løste dette problemet ved å bare velge en av de verdiene.

3.2.1 Brutekraft og uniform distribusjon

Vektoren y blir fylt opp med verdier fra x vektoren ved bruk av ligning (9). Når alle disse verdiene har blitt lagret så blir de sendt funksjonen *integrate*. Denne funksjonen bruker disse innsendte variabelene til å regne ut $ledd1$ som er eksponentielle dellen av integranden og $ledd2$ som er nevneren. Og dette gjentas i løkken n-ganger.

3.2.2 Forbedret MC og Importance sampling

I denne metoden er det flere variabler å ta hensyn til. r_i, $\cos(\theta)_i$ og ϕ_i blir regnet ved x_i-ene. Deretter så gjøres det samme som ovenfor. Disse sendes til funksjonen $ingerate - impor$ og her blir det regnet ut de forskjellige leddene i integranden. Dette gjøres n-ganger.

4 Resultater

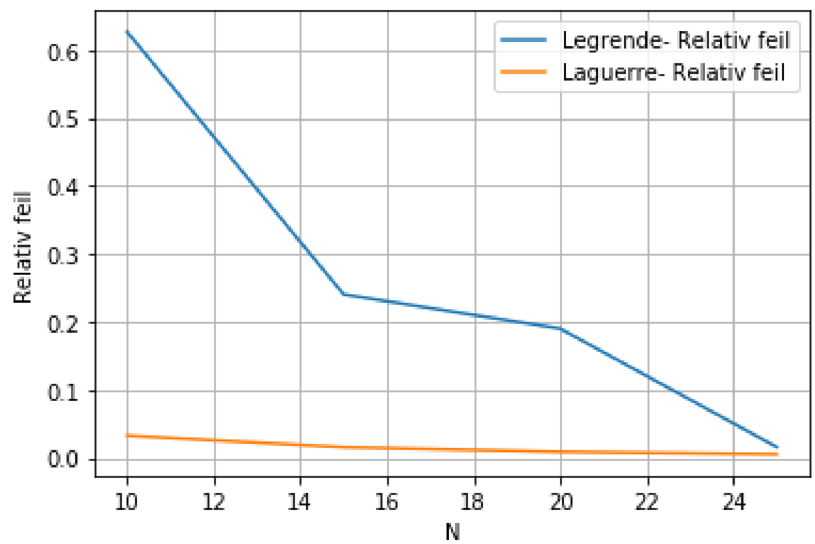
4.1 Gauss Kvadratur

Resultatene fra Gauss kvadraturet kan ses i tabell 1.

N	Svar-Legendre	Svar-Laguerre	Relativ Feil-Legendre	Relativ Feil-Laguerre
10	0.0720	0.1865	0.6266	0.0327
15	0.2391	0.1898	0.2403	0.0156
20	0.1561	0.1911	0.1900	0.0087
25	0.1958	0.1917	0.0158	0.0053

Tabell 1: Resultatene fra Gauss kvadraturet. N= antall integrasjonssteg, mens andre og tredje kolonne gir de approksimerte svarene til integralet. De to siste kolonnene gir relativ feil sammenliknet med det korrekte svaret: 0,1928.

Plottet over de relative feilene mot N kan ses i figur 2.



Figur 2: Relativ feil plottet for Legendre og Laguerre mot integrasjonssteglengde N. Feilene er de samme verdiene som fra tabell 1, men presentert på en annen måte da noen kan foretrekke plots fremfor grafer.

4.2 Monte Carlo

N	A (Brute force)	A (Importance)	Error (Brute force)	Error(Importance)
10e3	0.230631	0.183461	0.196433	0.04827
10e4	0.173337	0.194568	0.10079	0.00934
10e5	0.200374	0.193876	0.0394709	0.00575762
10e6	0.18113	0.192585	0.0603619	0.000937986

Tabell 2: Resultatene fra Monte Carlo. N= antall integrasjonssteg, mens andre og tredje kolonne gir de approksimerte svarene til integralet. De to siste kolonnene gir relativ feil sammenliknet med det korrekte svaret: 0,1928.

N	V (Brute force)	V(Importance)	Tid (Brute force)	Tid (Importance)
10e3	0.092088	0.00647	0.045208s	0.05656 s
10e4	0.0758239	0.00318618	0.509782 s	0.577573 s
10e5	0.0249523	0.000825743	3.11723 s	5.63679 s
10e6	0.00819576	0.000259464	34.7541 s	47.2875 s

Tabell 3: Resultatene fra Monte Carlo. N= antall integrasjonssteg, mens andre og tredje kolonne gir de approksimerte verdiene til variansene. De to siste kolonnene gir CPU tiden for algoritmen.

N	Tid (Parallisert)	Tid (Ikke parallisert)
10e3	0.0143 s	0.05656 s
10e4	0.100 s	0.577573
10e5	0.616 s	5.63679 s
10e6	3.45 s	47.2875 s

Tabell 4: Resultatene fra Monte Carlo med Important Sampling. N= antall integrasjonsprøver, andre kollonnen gir tiden etter parallisert programmet og tredje kollonnen gir tiden uten parallisering .

5 Diskusjon

Ved bruk av Gauss-Legendres metode gir denne ut ifra tabell 1 for høy error ved bruk av lave N. Det er ikke før N nærmer seg 25 at metoden gir tilnærminger som er brukbare. Det positive er at ut ifra figur 2, synker den relative erroren ganske raskt med økende N.

Når det gjelder Gauss-Laguerres metode synker feilen med økende N. Denne metoden gir også mye bedre approksimasjoner enn Legendre. Ved N=15 gir Laguerres metode tilsvarende relativ feil som Legendre hadde ved N=25.

Monte Carlo metoden fungerer best ved store N-verdier. Vi ser fra tabellene 2 og 3 at ved økende N-verdier så genererer algoritmen mer og mer presis resualtater. Samtidig kan det observeres at variansen minker mer og mer for økende N-verdier. Dette stemmer bra med teorien, nemlig at $\sigma_N = \frac{1}{\sqrt{N}}$. Variansen er omvendt proposjonal av kvadratroten til N-antall prøver. Jo større N-verdier desto mindre varianser og samtidig mindre relativ feil. Et annet interessant faktum er at variansen blir mindre ved modifisere metoden fra Brute-kraft til Importance Sampling, og dermed gir mer presis integral. Siden det kreves store N-verdier for å kjøre denne metoden så øker også CPU-tiden det tar for å kjøre programmet. Dette kan observeres i tabell 3. Det tar mer tid å kjøre metoden for Importance Sampling enn Brute-kraft. Dette skyldes at i Importance Sampling tas det hensyn til flere variabler og det gjør at maskinen bruker mer tid, men i Brute-kraft så er det nesten rett fram. Her tas det ikke hensyn til noen spesielle variabler men kun noen tilfeldige verdier.

6 Konklusjon

Det ble altså vist hvordan det er mulig å finne energien heliumatomet i gruntilstanden ved bruk av forskjellige integrasjonsmetoder. Det ble vist at den parallelliserte Monte Carlo metoden er den best tilpassede metoden å velge til et slikt problem da denne både gir en gode approksimasjoner og er tidseffektiv.Det ble testet to former for Monte Carlo metoden. I den ene delen så ble det antatt uniform distribusjon og i den andre delen ble det antatt eksponentiell distribusjon. Det viste seg at vi får mer presise resualtater, og mindre varianser, i den forbedrede versjonen av Monte Carlo metoden. Det ble også vist at Gauss-Laguerre foretrekkes fremfor Gauss-Legendre da Laguerre gir bedre tilnærminger, samtidig som den er enkel å bruke ved integrasjonsgrenser som går mot ∞.

7 Appendiks

7.1 Omskrivning til sfæriske koordinater og "Laguerres form"

For å skrive om til sfæriske koordinater tas det utgangspunkt i integralet som ble oppgitt, der α=2.

$$A = \int_{-\infty}^{\infty} dr_1 dr_2 e^{-4(r_1+r_2)} \frac{1}{|r_1 - r_2|} \tag{11}$$

Hvor $dr_1 dr_2 = r_1^2 r_2^2 dr_1 dr_2 dcos(\theta_1) dcos(\theta_2) d\phi_1 d\phi_2$
som igjen gir $dr_1 dr_2 = r_1^2 r_2^2 \sin\theta_1 \sin\theta_2 dr_1 dr_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2$.

Dette føres i likning 11 og gir:

$$A = \int_0^{\infty}\int_0^{\infty}\int_0^{\pi}\int_0^{\pi}\int_0^{2\pi}\int_0^{2\pi} r_1^2 r_2^2 e^{-4r_1} e^{-4r_2} \frac{\sin\theta_1 \sin\theta_2}{r_{12}} dr_1 dr_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2 \tag{12}$$

Så gjøres uttrykket om ved å forandre variabler: $u_1 = 4r_1$ og $u_2 = 4r_2$, dette gir:

$$A = \int_0^{\infty}\int_0^{\infty}\int_0^{\pi}\int_0^{\pi}\int_0^{2\pi}\int_0^{2\pi} \frac{1}{(2\alpha)^5} u_1^2 u_2^2 e^{-u_1} e^{-u_2} \frac{\sin\theta_1 \sin\theta_2}{\sqrt{u_1^2 + u_2^2 - 2u_1 r_2 \cos\beta}} du_1 du_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2 \tag{13}$$

Som gir integralet vårt på "Laguerres form"

$$A = \frac{1}{(2\alpha)^5}\int_0^{\infty}\int_0^{\infty}\int_0^{\pi}\int_0^{\pi}\int_0^{2\pi}\int_0^{2\pi} W(u_1)W(u_2)g(u,\theta,\phi)\, du_1 du_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2. \tag{14}$$

7.2 Monte Carlo

7.2.1 Utledning av variance

Variansen er definert ved:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - \langle f \rangle)^2 p(x_i)$$

Der $\langle f \rangle = \sum_{i=1}^{N}(f(x_i)p(x_i)$ og $p(x_i)$ er distribusjonsfunksjonen. Vi kan da skrive som:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(f(x_i)p(x_i))^2 - (\frac{1}{N}\sum_{i=1}^{N}(f(x_i)p(x_i)))^2$$

Som kan igjen skrives som:

$$\sigma_N^2 = \frac{1}{N}(\langle f^2\rangle - \langle f^2\rangle)$$

7.2.2 Utledning av I med Brute-kraft

Vi ser på integralet på følgendevis:

$$A = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(x1, y1, z1, x2, y2, z2) dx_1 dy_1 dz_1 dx_2 dy_2 dz_2$$

Det kan vi approksimere til:

$$A = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(x1, y1, z1, x2, y2, z2) dx_1 dy_1 dz_1 dx_2 dy_2 dz_2 \approx J \frac{1}{N} \sum_{i=1}^{N} (f(x_i) p(x_i)$$

Der J er Jacobi-determinanten og kan beregnes:

$$J = \Pi_{i=1}^{6}(b_i - a_i) = (b_i - a_i)^6$$

Så vi har da:

$$A \approx (b_i - a_i)^6 \frac{1}{N} \sum_{i=1}^{N} (f(x_i) p(x_i)$$

7.2.3 Utledning av uniforme distributerte tilfeldige tall

Antar at vi har en generel uniform distribusjon:

$$p(y)dy = \begin{cases} \frac{dy}{b-a}, y \in [a,b] \\ \\ 0, ellers \end{cases}$$

Vi ønsker å relatere dette til x-verdiene, $x \in [0, 1]$:

$$p(y)dy = \frac{dy}{b-a} = dx p(x) = dx$$

Ved integrasjon:

$$\int_0^x dx' = \int_a^y \frac{dy'}{b-a}$$

Altså:

$$x = \frac{y}{b-a} - \frac{a}{b-a}$$

Dersom vi får y til å stå alene:

$$y = a + (b-a)x$$

For å ikke gjøre dette forvirrende med variabler så kalles denne s_i:

$$s_i = a + (b-a)x_i$$

Det er s-verdiene som inngår i funksjonen for å beregne integralet med Brute-kraft.

7.2.4 Utledning av sfæriske koordinater

For Importance Sampling ble det valgt en eksponentiell distribusjonsfunksjon og deretter byttet om til sfæriske koordinater. Så vi har at:

$$p(r)dr = dr4e^{-r4} = dxp(x) = dx$$

Integrer på begge sidene:

$$x(r) = P(r) = \int_0^r 4e^{-4r}dr' = 1 - e^{-4r}$$

Skriver det som funksjon av r_i:

$$r_i = \frac{1}{4}\ln(1 - x_i),$$

For å gjøre det enklere så ser vi på $\cos(\theta)_i$ som variabel og ikke θ_i. Siden $\theta_i \in [0, \pi]$ så blir $\cos(\theta_i) \in [-1, 1]$. Derfor kan $\cos(\theta_i)$ fås ved:

$$\cos(\theta)_i = 2x_i - 1$$

Til slutt har vi ϕ-ene. Siden $\phi \in [0, 2\pi]$ så kan det skrives på denne måten:

$$\phi_i = 2\pi x_i$$

Der $x_i \in [0, 1]$.

7.2.5 Utledning av I med Importance Sampling

Vi ser på integralet på følgendevis:

$$A = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(x1, y1, z1, x2, y2, z2)dx_1dy_1dz_1dx_2dy_2dz_2$$

Der

$$f(x1, y1, z1, x2, y2, z2) = e^{-4(r_1+r_2)}\frac{1}{|r_1 - r_2|}$$

Ved å transformere fra kartesisk til sfæriske koordinater får vi:

$$A = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(x1, y1, z1, x2, y2, z2)dx_1dy_1dz_1dx_2dy_2dz_2$$

$$= \int_0^{\infty}\int_0^{\infty}\int_0^{\pi}\int_0^{\pi}\int_0^{2\pi}\int_0^{2\pi} e^{-4r_1}e^{-4r_2}\frac{1}{r_{12}}r_1^2r_2^2dr_1dr_2dcos(\theta_1)dcos(\theta_2)d\phi_1d\phi_2$$

Dette gir den nye funksjonen:

$$f(r_{1i}, r_{2i}, \cos(\theta)_{1i}, \cos(\theta)_{2i}, \phi_{1i}, \phi_{2i}) = r_1^2r_2^2e^{-4r_1}e^{-4r_2}\frac{1}{r_{12}}$$

Vi kan skrive integralet som:

$$A = \int_0^\infty \int_0^\infty \int_0^\pi \int_0^\pi \int_0^{2\pi} \int_0^{2\pi} \frac{f(r_{1i}, r_{2i}, \cos(\theta)_{1i}, \cos(\theta)_{2i}, \phi_{1i}, \phi_{2i})}{p(r)} p(r)...$$

$$...dr_1 dr_2 dcos(\theta_1) dcos(\theta_2) d\phi_1 d\phi_2$$

$$A = \int_0^\infty \int_0^\infty \int_0^\pi \int_0^\pi \int_0^{2\pi} \int_0^{2\pi} \frac{f(r_1(x_i), r_2(x_i), \cos(\theta)_1(x_i), \cos(\theta)_2(x_i), \phi_1(x_i), \phi_2(x_i))}{p(r(x_i))}...$$

$$...dr_1 dr_2 dcos(\theta_1) dcos(\theta_2) d\phi_1 d\phi_2$$

Integralet kan dermed approksimeres til:

$$A \approx J \frac{1}{N} \sum_{i=1}^{N} \frac{f(r_1(x_i), r_2(x_i), \cos(\theta)_1(x_i), \cos(\theta)_2(x_i), \phi_1(x_i), \phi_2(x_i))}{p(r(x_i))}$$

$$= J \frac{1}{N} \sum_{i=1}^{N} \frac{r_1^2 r_2^2}{r_{12}}$$

Der $J = (2\pi)^2 2^2 \frac{1}{4^2}$ er Jacobi determinanten.

8 Bibliografi

[1] Morten Hjorth-Jensen, august 2015, Computational Physics Lecture Notes, 7 Eigensystems, (https://github.com/CompPhysics/ComputationalPhysics/ blob/master/doc/Lectures/lectures20

[2] Morten Hjorth-Jensen, 2019, Codeexamples, (https://github.com/CompPhysics/ComputationalPhysics/tree/master/doc/Projects/2019/Project3/CodeExam

[3] Morten Hjorth-Jensen, september 2019 Project 3, https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects /2019/Project3/pdf/Project3.pdf.

Link til githubmappen: https://github.com/philipkarim/Philip-and-Rohullah-ComFys