

FYS3150 Computational Physics 2014

## Oblig 2

**Løysning av Schrödinger likninga for to elektron i ein tredimensjonal oscillator brønn.**

Øyvind Sigmundson Schøyen

28. september 2014

### **Sammendrag**

Skriv eit samandrag av prosjektet her oppe.  
<https://github.com/Schoyen/FYS3150/tree/master/Oblig2>

# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>4</b>
<b>2</b>	<b>Jacobirotasjon</b>	<b>4</b>
2.1	Algoritma . . . . .	4
2.2	Implementering . . . . .	6
<b>3</b>	<b>Resultat</b>	<b>6</b>
3.1	Jacobi . . . . .	6
3.2	Potensialenergien til elektron med og uten Coulomb interaksjon	7
3.2.1	Plott for $\omega_r = 0.01$ . . . . .	8
3.2.2	Plott for $\omega_r = 0.5$ . . . . .	9
3.2.3	Plott for $\omega_r = 1$ . . . . .	10
3.2.4	Plott for $\omega_r = 5$ . . . . .	13
3.3	Feilanalyse . . . . .	13
3.3.1	Plott for $\rho_{\max} = 2$ . . . . .	14
3.3.2	Plott for $\rho_{\max} = 3$ . . . . .	15
3.3.3	Plott for $\rho_{\max} = 4$ . . . . .	15
3.3.4	Plott for $\rho_{\max} = 5$ . . . . .	16
3.3.5	Plott for $\rho_{\max} = 7$ . . . . .	16
3.3.6	Plott for $\rho_{\max} = 10$ . . . . .	17

## 1 Introduksjon

I dette prosjektet er me interesserte i å løyse Schrödinger likninga for to elektron. Likninga er skriva om slik at me kan jobbe med eit ein-lekam problem istadenfor to. Me nyttar lineær algebra for å løyse differensiallikningane som eit sett med lineær likningar. Måten me gjer dette på er ved Jacobirotasjon for å finne eigenvektorar og eigenverdiar. Til slutt vil me plotte bølgefunksjonen for grunntilstanden til elektrona ved hjelp av eigenvektorane og eigenverdiane.

## 2 Jacobirotasjon

For å løyse eigenverdi- og eigenvektorproblem vil me nytte Jacobirotasjon. Dette er ein algoritme som, etter ein rekke similaritetsformasjonar, vil gjere alle ikkje-diagonale matriseelement til null. Denne algoritmen er likevel ikkje ein veldig effektiv algoritme då me ved ein rotasjon kan kome i skade for å gjere eit element som tidlegare var null til å bli ikkje-null. Numerisk kan det og ta lang tid før elementa vert null. Me vil difor heile tida teste verdiane mot ein toleranse.

### 2.1 Algoritma

Ein similaritetstransformasjon er gitt ved

$$B = S^T A S$$

kor  $S$  er ein ortogonal matrise der  $SS^T = SS^{-1} = I$ . Matrisa  $S$  transformerer  $A$  ein vinkel  $\theta$  i planet medan  $S^T$  tek ho tilbake. Me vil då velje  $\theta$  slik at alle ikkje-diagonale element vert null. Når me gjer dette numerisk må me gjere ein rekke similaritetstransformasjonar for å oppnå dette. Då har me

$$B = S_n^T \dots S_1^T A S_1 \dots S_n.$$

Kvar matrise  $S$  og  $S^T$  er identitetsmatrisa med unntak av elementa  $s_{kk} = s_{ll} = \cos \theta$ ,  $s_{kl} = -s_{lk} = -\sin \theta$  og  $s_{ii} = 1$  for  $i \neq k$  og  $i \neq l$ . Produktet  $B = S^T A S$  kan då skrivast som

$$\begin{aligned} b_{ii} &= a_{ii}, & i \neq k, i \neq l \\ b_{ik} &= a_{ik} \cos \theta - a_{il} \sin \theta, & i \neq k, i \neq l \\ b_{il} &= a_{il} \cos \theta + a_{ik} \sin \theta, & i \neq k, i \neq l \\ b_{kk} &= a_{kk} \cos^2 \theta - 2a_{kl} \cos \theta \sin \theta + a_{ll} \sin^2 \theta \\ b_{ll} &= a_{ll} \cos^2 \theta + 2a_{kl} \cos \theta \sin \theta + a_{kk} \sin^2 \theta \\ b_{kl} &= (a_{kk} - a_{ll}) \cos \theta \sin \theta + a_{kl}(\cos^2 \theta - \sin^2 \theta). \end{aligned}$$

Me vil no velje  $\theta$  slik at alle ikkje-diagonale element  $b_{kl}$  i praksis vert null. For kvar iterasjon vil me då teste om summen av alle dei ikkje-diagonale elementa er mindre enn ein toleranse  $\epsilon$ . Me vil derimot ikkje gjer dette då det er ein tidkrevande prosess. Erstatninga vert då å sjå om det største elementet blant dei ikkje-diagonale elementa er mindre enn  $\epsilon$ . Dette vil då vere

$$|a_{kl}| = \max_{i \neq j} |a_{ij}|.$$

Me krever at  $b_{kl} = b_{lk} = 0$ . Det gjer oss likninga

$$a_{kl}(c^2 - s^2) + (a_{kk} - a_{ll})cs = b_{kl} = 0. \quad (1)$$

Me definerer no

$$\tau = \frac{a_{ll} - a_{kk}}{2a_{kl}} \quad \Rightarrow \quad a_{ll} - a_{kk} = 2\tau a_{kl}.$$

Setter dette inn i (1) og får

$$\begin{aligned} a_{kl}c^2 - a_{kl}s^2 - 2\tau a_{kl}cs &= 0, \\ \Rightarrow \quad a_{kl} - a_{kl}\frac{s^2}{c^2} - 2\tau a_{kl}\frac{s}{c} &= 0. \end{aligned}$$

Siden  $c = \cos \theta$  og  $s = \sin \theta$  vil me få

$$\begin{aligned} a_{kl} - a_{kl} \tan^2 \theta - 2\tau a_{kl} \tan \theta &= 0, \\ \Rightarrow \quad 1 - t^2 - 2\tau t &= 0, \\ \Rightarrow \quad t^2 + 2\tau t - 1 &= 0, \end{aligned}$$

kor  $t = \tan \theta$ . Me vil då få

$$\begin{aligned} t &= \frac{-2\tau \pm \sqrt{4\tau^2 - 4(-1)}}{2} \\ &= -\tau \pm \sqrt{1 + \tau^2}. \end{aligned}$$

For å unngå avrundingsfeil ved  $\tau \gg 0$  gonger me andregradslikninga med den konjugerte. Det vil gje oss

$$\begin{aligned} t &= \left(-\tau \pm \sqrt{1 + \tau^2}\right) \left(\frac{-\tau \pm \sqrt{1 + \tau^2}}{-\tau \pm \sqrt{1 + \tau^2}}\right) \\ &= \frac{-\tau^2 + (1 + \tau^2)}{\tau \pm \sqrt{1 + \tau^2}} = \frac{1}{\tau \pm \sqrt{1 + \tau^2}}, \end{aligned}$$

som gjer oss

$$t = \frac{1}{\tau \pm \sqrt{1 + \tau^2}} \quad \vee \quad t = \frac{-1}{-\tau + \sqrt{1 + \tau^2}}.$$

Me vil no velje den minste av røttene  $t$  slik at  $c$  og  $s$  henholdsvis går mot ein og null ved

$$c = \frac{1}{\sqrt{1 + t^2}}, \quad s = tc.$$

Då vil me kunne begrense vinkelen  $\theta \leq \frac{\pi}{4}$  slik at differansen mellom matrisa  $B$  og  $A$  vert så liten som mogleg ved formelen

$$|B - A|_F^2 = 4 \underbrace{(1 - c)}_{=0 \text{ for } c \rightarrow 1} \sum_{i=1, i \neq k, l}^n (a_{ik}^2 + a_{il}^2) + \frac{2a_{kl}^2}{c^2}.$$

Me vil fortsette desse operasjonane heilt til  $\max(a_{ij})^2 \leq \epsilon$  for  $i \neq j$ .

## 2.2 Implementering

Algoritma vert implementert i ein klasse med tre metodar som løyser eigenverdi og eigenvektor problemet.

## 3 Resultat

Denne seksjonen vil bli delt opp i tre deler. Me vil diskutere metoden i seg sjølv kor me ser på køyretid og stabilitet. Me vil sjå på energinivåa til elektronen med, og uten, Coulomb interaksjon for forskjellige  $\omega_r$  og  $\rho_{\max}$ . Eg har valt å gjere dette ved Jacobi. Dette krev at me normaliserer eigenvektorane for å få riktige verdiar. Det vil bli forklart i meir nøyaktighet. Til slutt har eg lagt ved ein feilanalyse på eigenverdiane som funksjon av  $n_{\text{step}}$ .

### 3.1 Jacobi

Jacobirotasjon er ein reknetung algoritme. Denne metoda bryt veldig kjapt ned når ein lager ei stor matrise. For vårt formål er difor essensielt at me uttnyttar peikar- og referansemoglegheitane i C++ for å unngå unødvendig tidbruk på lagring. Køyretida til metoden går som  $\mathcal{O}(n^3)$  for rotasjon. Mykje av grunnen til at metoda er treig kjem frå det faktum at me står i fare for å gjere eit element som er null til ikkje-null ved ein rotasjon. Me merker fort når matrisa blir stor at tidbruken stig veldig kjapt. Ein samanlikning av `eig_sym` frå Armadillo og Jacobi gjer oss køyretidene i sekund i tabellen under.

Steg	Armadillo	Jacobi	Antal iterasjoner
n = 10	0.0987122	7.1168e-05	128
n = 25	0.000693803	0.00148126	879
n = 50	0.00604665	0.016172	3682
n = 75	0.00899465	0.0759956	8472
n = 100	0.0153569	0.228282	15107
n = 150	0.0275323	1.16203	34296
n = 200	0.0574736	3.47677	61181
n = 250	0.0954146	8.4202	95818

Me ser korleis Armadillo aukar jamnt med  $n$ -verdiane medan Jacobi stig eksponensielt. Med ei kubisk køyretid vil me veldig kjapt nå eit tak over kor stor matrisa vår kan vere. I programmet vårt set me ei begrensning for kor mange iterasjonar me tillet. Når matrisa vert stor nok vil denne begrensninga ikkje lengre tjene formålet då ho vil verte for stor for våre praktiske formål. Ei senkning av denne kan igjen føre til eit for lite antal iterasjoner og gå på bekostning av presisjon. Antal iterasjonar som står oppførte er det antalet det tek før alle ikkje-diagonale matriseelement i praksis er null. Metoden i seg sjølv er difor stabil for mindre matrisar, men kan fort bli ubrukeleg for større matrisar. Fordelane med metoda er at ho er enkel å implementere og er effektiv ved parallellprogrammering. Me kjem diverre ikkje inn på dette i denne oppgåva.

Alle plott er plotta for  $n_{\text{step}} = 200$ .

### 3.2 Potensialenergien til elektron med og uten Coulomb interaksjon

I dette prosjektet har me rekna på potensialenergien til to elektron i ein oscillator brønn. Fyrst har me sett på eit døme uten Coulomb interaksjon. Då har me definert det harmoniske oscillator potensialet ved  $V_i = \rho_i^2$ . For dette dømet har me eksakte løysningar på eigenverdiane me får frå Jacobi. Dette avheng derimot av val av  $n_{\text{step}}$  og  $\rho_{\text{max}}$ . For å få eit svar med fire siffrs nøyaktighet må me inngå nokre kompromiss for  $\rho_{\text{max}}$ . Dette kjem me tilbake til i feilanalyse. Når me ser på potensialenergien med Coulomb interaksjon vil me får eit potensial gitt ved  $V_i = \omega_r^2 \rho^2 + \frac{1}{\rho}$ . Her vil me fortsatt vere avhengige av  $n_{\text{step}}$  og  $\rho_{\text{max}}$ , men me vil no i tillegg ha ein avhengighet til  $\omega_r$ . Valet av  $\rho_{\text{max}}$  vil vere direkte knytta til  $\omega_r$ . Me nyttar artikkelen til M. Taut slik at me kan finne verdier med ein eksakt løysning. Diverre vil kun nokre få av desse løysningane vere relevante for vårt formål.

Me er interesserte i  $\omega_r = 0.01$ ,  $\omega_r = 0.5$ ,  $\omega_r = 1$  og  $\omega_r = 5$ . Artikkelen til M. Taut hjelper oss med to av desse. Me kan finne ei eksakt løysning for  $\omega_r = 0.05$  og  $\omega_r = 0.25$ . Me finn då ein  $\rho_{\text{max}}$  som gjer oss eksakte svar og seier at for  $\omega_r = 0.01$  og  $\omega_r = 0.5$  så vert det tilnærma likt. Me kjem då fram til at frekvensane  $\omega_r = 0.01$  og  $\omega_r = 0.5$  vil gje oss  $\rho_{\text{max}} \approx 50$  og  $\rho_{\text{max}} \approx 10$ . Desse frekvensane tilsvarer svak Coulomb interaksjon og me vil få “flate kurver”.

For  $\omega_r = 1$  vil me velje same  $\rho_{\max}$  som for det tidlegare potensialet. Me vil då sjå at leddet  $\frac{1}{\rho}$  utgjer ein forskyvning av bølgefunksjonen. Det gjer oss  $\rho_{\max} = 5$ . For den siste frekvensen  $\omega_r = 5$  vil me ha ein liten  $\rho_{\max}$  då dette utgjer ein sterk Coulomb interaksjon. Bølgefunksjonen vil då bli bratt og kort. Me vel  $\rho_{\max} = 2$ .

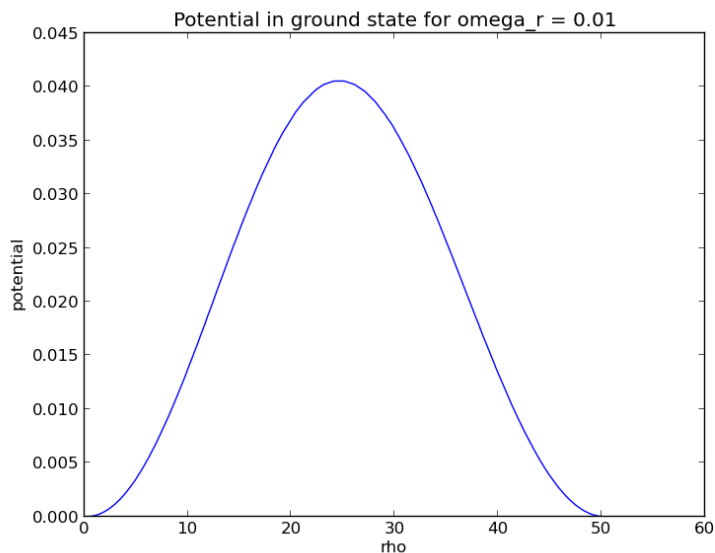
Siden me behandlar alle verdiar som dimensjonslause er me interesserte i å plote normaliserte eigenvektorar. Det gjer me ved formelen

$$\mathbf{v} = \frac{\mathbf{v}}{\sqrt{\int_{\rho_0}^{\rho_{\max}} v^2 d\rho}}.$$

Numerisk nyttar me trapesregelen for å rekne ut integralet. Alle grafane vil difor kun spenne ut eit areal på 1.

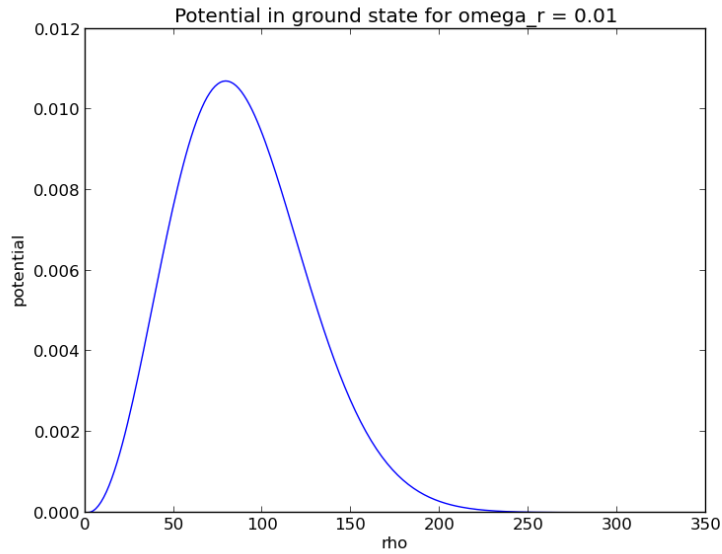
### 3.2.1 Plott for $\omega_r = 0.01$ .

Sjølv om den eksakte løysninga for  $\omega_r = 0.05$  vil gje oss ein  $\rho_{\max} \approx 50$  vil me for  $\omega_r = 0.01$  ha ein  $\rho_{\max}$  som er mykje større. For å få eit plott som ikkje blir “trykt” ned mot slutten på grunn av randkrava har eg plotta for  $\rho_{\max} = 300$  i tillegg. Diverre resulterer dette i eit lite informativt plott når me plottar for alle frekvensane i lag.



Figur 1: Her har me plotta for  $\rho_{\max} = 50$ . Me ser korleis kurva vert avkappa mot slutten grunna randkrava.

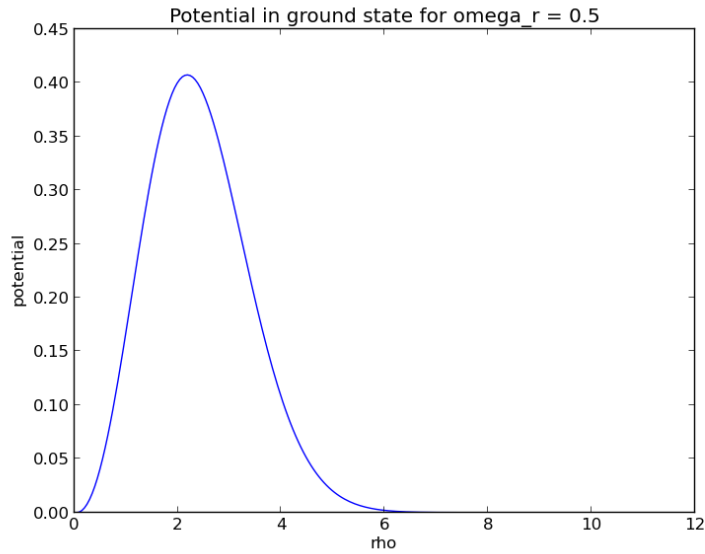




Figur 2: Her har me plotta for  $\rho_{\max} = 300$ . I dette dømet vil ikkje kurva vere begrensa av randkrava. Me kan sjå på verdiane til potensialet at kurva vil vere mykje mindre enn dei for kortare  $\rho_{\max}$ . Dette då arealet under alle grafane vil vere 1.

### 3.2.2 Plott for $\omega_r = 0.5$ .

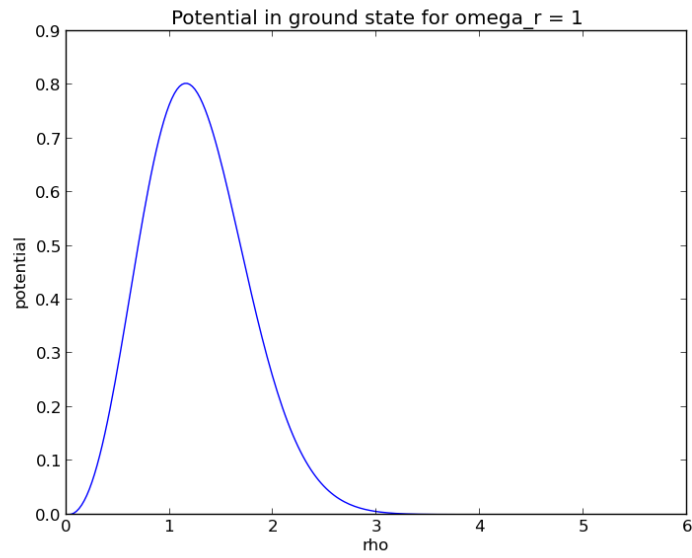
Her vil me ha ei tilnærma løysning frå artikkelen til M. Taut.



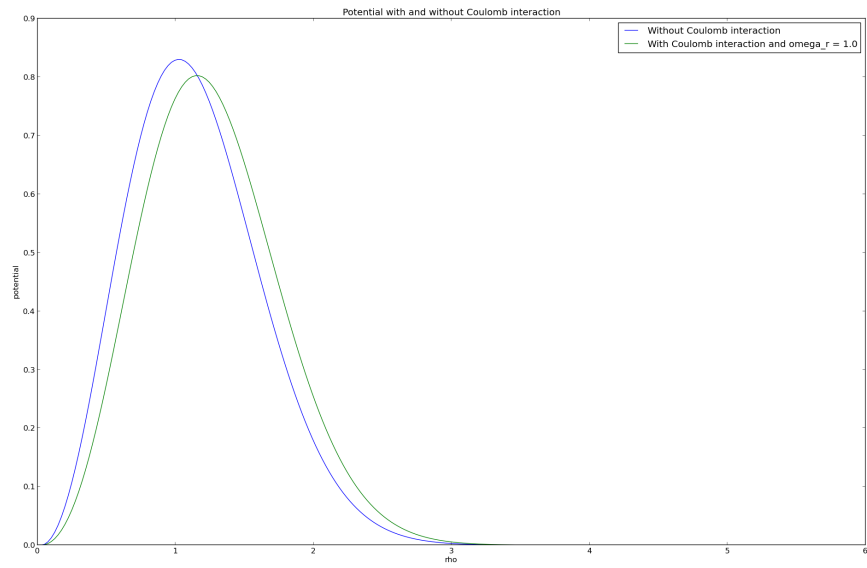
Figur 3: Plott for  $\rho_{\max} = 10$ . Me ser at kurva ikkje vert kappa av frå randkrava. Legg merke med aukninga i potensialet. Arealet vil fortsatt vere 1, men siden  $\rho_{\max}$  er mykje mindre enn for det forrige plottet vil verdien stige.

### 3.2.3 Plott for $\omega_r = 1$ .

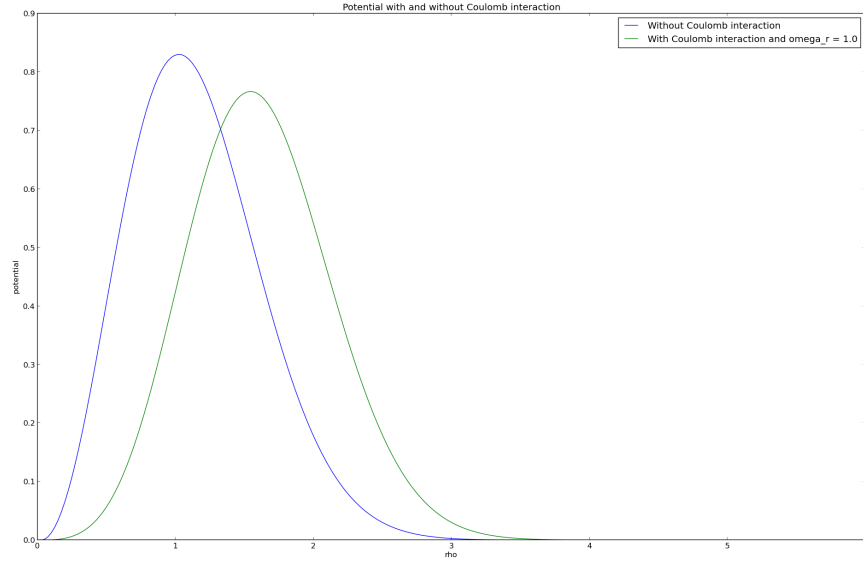
I dette plottet vil det vere hensiktsmessig å ha med samanlikning av det føregåande og nåværande potensialet. Dette då me vil kunne sjå ei forskyvning. Viss me aukar  $\frac{C}{\rho}$ -leddet i potensialet med Coulomb interaksjon vil me kunne sjå ei tydeleg forskyvning.



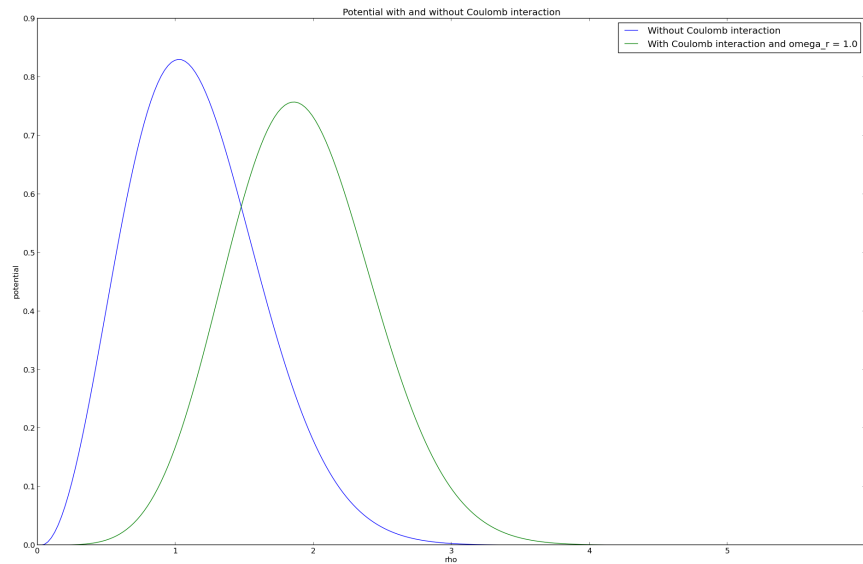
Figur 4: Plott for  $\rho_{\max} = 5$ .



Figur 5: Her er  $V_i = \omega_r^2 \rho^2 + \frac{1}{\rho}$  for den grøne grafen.



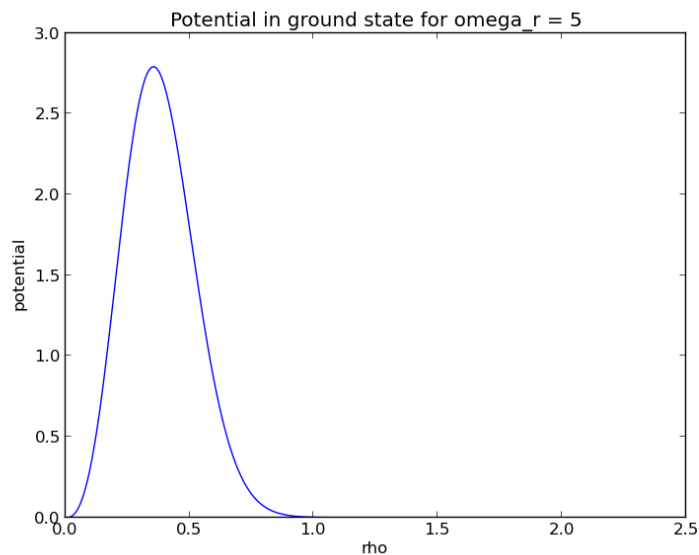
Figur 6: Her er  $V_i = \omega_r^2 \rho^2 + \frac{5}{\rho}$  for den grøne grafen.



Figur 7: Her er  $V_i = \omega_r^2 \rho^2 + \frac{10}{\rho}$  for den grøne grafen.

### 3.2.4 Plott for $\omega_r = 5$

Den siste frekvensen vil gje oss ein bratt, men kort graf. Me får eit kraftig potensial for elektron i nærleiken av einannan. Her har me  $\rho_{\max} = 2$ .

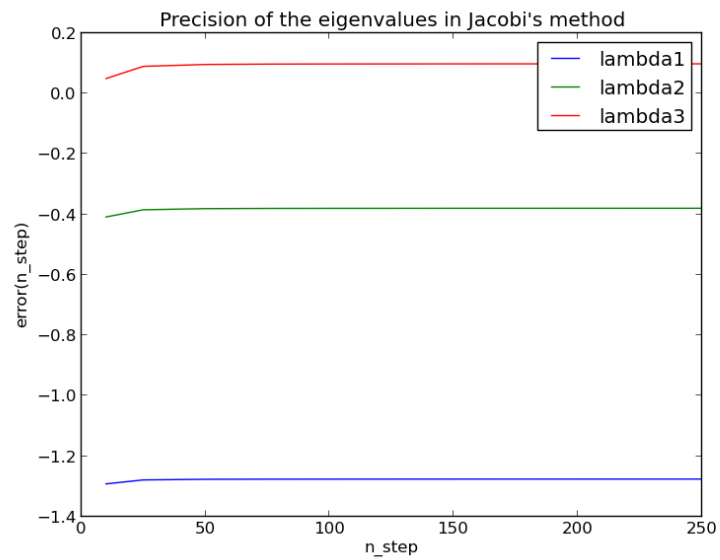


Figur 8: Plott for  $\rho_{\max} = 2$  gjer oss ei kurve med eit stort potensial.

### 3.3 Feilanalyse

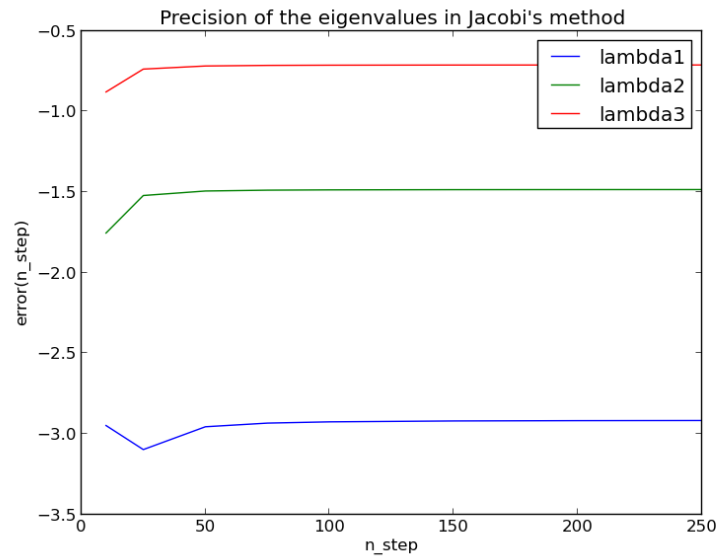
Eg valt å inkludere ei lita feilanalyse for å få eit overblikk over kor lang tid det tek før Jacobis metode gjer oss fire siffers nøyaktighet (merk at det er her snakk om siffer og ikkje desimal) på dei tre fyrste eigenverdiane. Diverre er ikkje dette eit enkelt svar då det avhenger av valet vårt av  $\rho_{\max}$ . Eg har her plotta  $\log_{10}$  av differansen mellom den eksakte og den numeriske løsyninga (og trukket frå ein då  $\log_{10}$  vil gje oss antal desimalar) mot  $n_{\text{step}}$ .

### 3.3.1 Plott for $\rho_{\max} = 2$



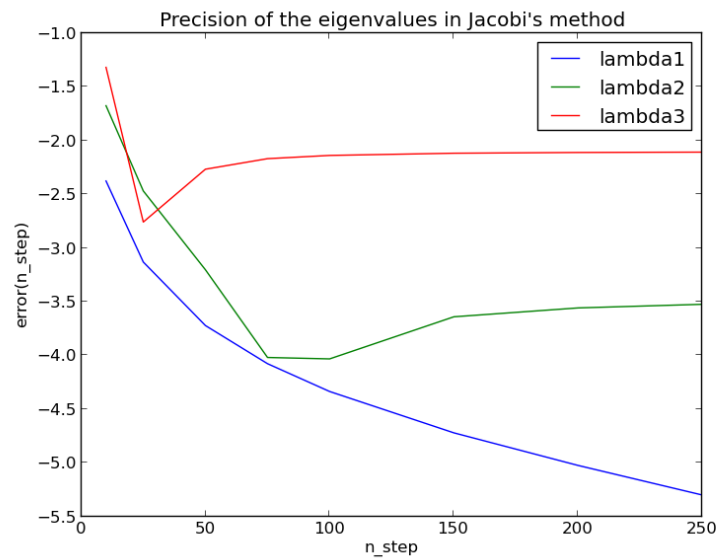
Figur 9: Uttrykket me jobber med for potensialet er berekna for store verdier av  $\rho_{\max}$ . Me får difor store feil for dei eksakte eigenverdiane.

### 3.3.2 Plott for $\rho_{\max} = 3$



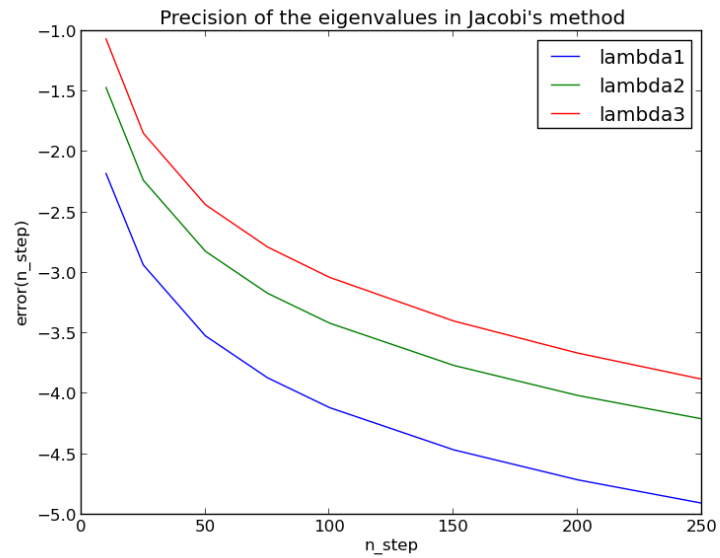
Figur 10: Her har me mykje den same problematikken. Metoden vil sikte seg inn mot feil eigenverdi.

### 3.3.3 Plott for $\rho_{\max} = 4$



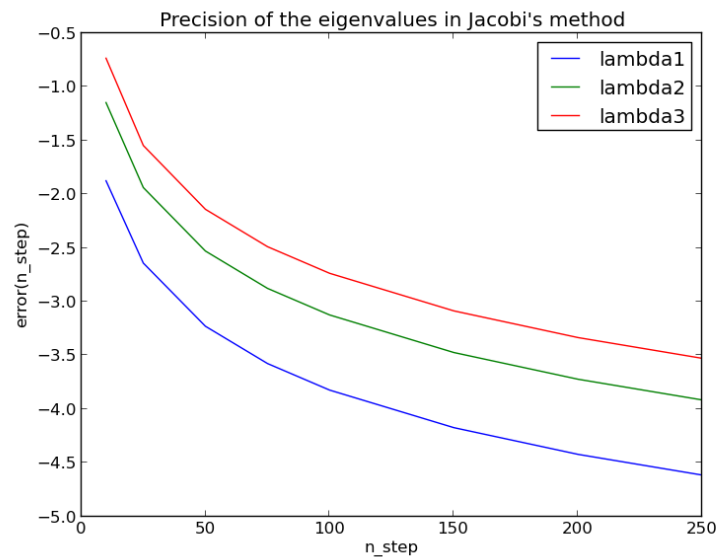
Figur 11: Her byrjar metoden å fungere som han skal.

### 3.3.4 Plott for $\rho_{\max} = 5$



Figur 12: For  $\rho_{\max} = 5$  får me oppførselen me vil ha.

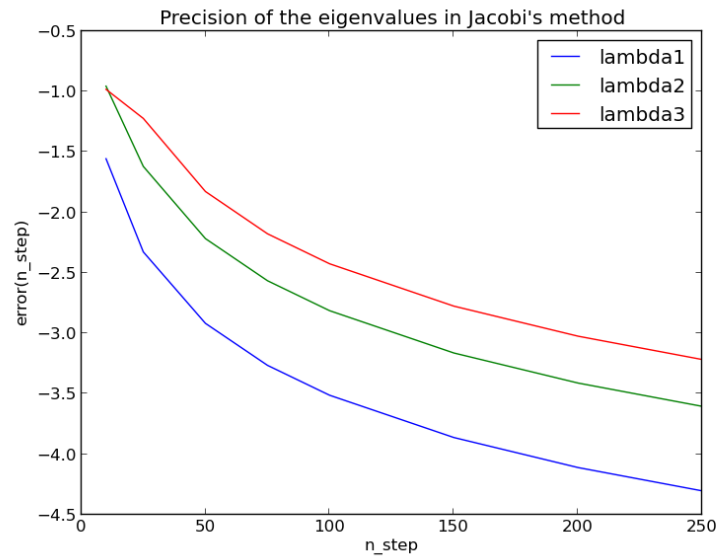
### 3.3.5 Plott for $\rho_{\max} = 7$



Figur 13: Her kan me sjå korleis feilen byrjar å stige igjen.



### 3.3.6 Plott for $\rho_{\max} = 10$



Figur 14: Gradvis stig feilen igjen.

For  $\rho_{\max} = 5$  får me dei verdiane me ynskjer. For den fyrste eigenverdien klarer me nesten å få fem siffers nøyaktighet. Viss me minkar eller minkar  $\rho_{\max}$  vil feilen stige igjen.