

Toteutusdokumentti: elektronirakenteen
optimointi atomiorbitaalittomaan
tiheysfunktionaaliteorian avulla

Markus KAUKONEN

21. kesäkuuta 2014

email: markus.kaukonen@iki.fi, opiskelijanumero: 010974524

systeemi	menetelmä	Cpu [s]	Energia [H]
pieni	MC	0.6	1174.54
keski	MC	27.6	3178.85
suuri	MC	535.2	329.98
pieni	SD	16.8	1174.47
keski	SD	338	3169.44
suuri	SD	26560	320.49

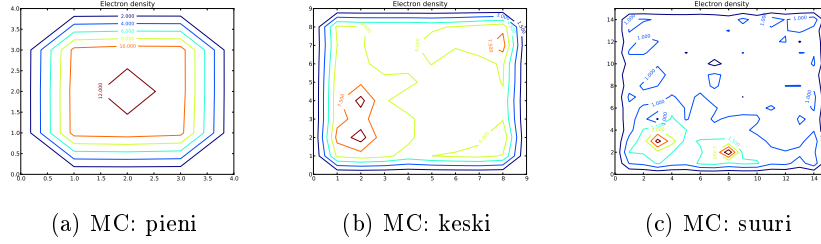
Taulukko 1: Verrataan monte carlo (MC) ja steepest descent SD minimointien kuluttamia Cpu-aikoja ja saatua energian minimiarvoa.

1 Ohjelman yleisrakenne

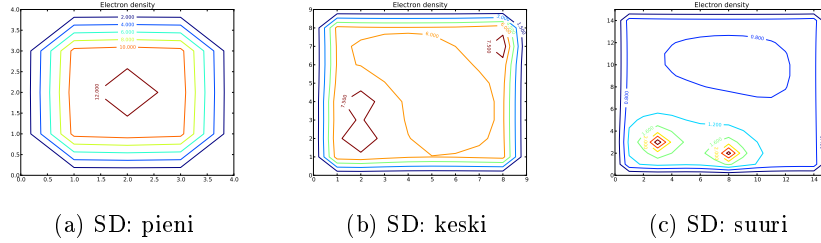
Ohjelma minimoi kokonaisenergiaa joko Monte Carlo (MC) tai Steepest Descent (SD) menetelmällä. Minimointimenetelmä on kuvattu määrittelydokumentissa. MC minimointiin liittyvät algoritmit ovat paketissa 'laskentaa.py'. SD minimointiin liittyvät algoritmit ovat paketissa 'laskentaa_sd.py'. Kokonaisenergia lasketaan määrittelydokumentissa kuvatulla tavalla. Energian laskemiseen liittyvät funktiot ovat paketissa 'energiat.py'. Samassa paketissa lasketaan myös elektronien aiheuttama Hartree potentiaali, joka iteroidaan Gauss-Seidel iteraatiolla (Gauss Seidel iteraatio sijaitsee paketissa 'laskentaa.py'). Alkuarvot luetaan tiedostoista jotka on nimetty pääohjelmassa 'vertaile_nopeuksia.py'. Luentarutiinit ovat pakkauksessa 'read_data.py'. Elektronitiheydet, Hartree potentiaali ja energian funktionaaliderivaatta tiheyden suhteen esitetään kolmi- tai kaksiulotteisissa grideissä. Gridiobjekti on määritelty paketissa 'gridi.py'. Tässä työssä pääohjelma 'vertaile_nopeuksia.py' tekee energian minimoinnin kolmelle eri 2d-systeemille (input tiedostot: 'alkuarvot.txt_5x5x0', 'alkuarvot.txt_10x10x0', 'alkuarvot.txt_16x16x0'). Minimointi tehdään sekä monte carlo menetelmällä että steepest descent menetelmällä. Lisäksi minimoidusta elektronitiheydestä piirretään kuva (piirtorutiineja on paketissa 'piirtoa.py').

2 Suorituskykyvertailu

Työssä tehtiin minimointia kahdella eri menetelmällä (MC ja SD). Allaolevassa taulukossa on esitetty minimoitu energia ja minimointiin kulunut aika systeemin koon funktiona. Laskenta tehtiin ACER aspire (vm. 2010) kannettavalla tietokoneella (Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz). Systeemien koot pienimmästä suurimpaan ovat 5x5, 10x10 ja 16x16 gridipistettä. Atomiytimet on sijoitettu kohtiin: 5x5: ([2,2, varaus +1]), 10x10: ([2,2,+1],[2,4,+1],[8,7,+2], 16x16: ([3,3,+1],[8,2,+1]).



Kuva 1: Monte Carlo (MC) simulointien tuottamia elektronitiheyksiä kolmessa testisysteemissä.

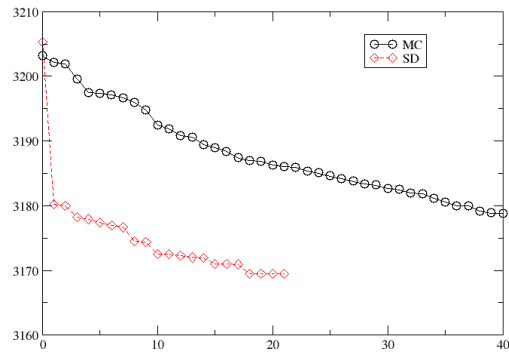


Kuva 2: Steepest Descent (SD) simulointien tuottamia elektronitiheyksiä kolmessa testisysteemissä.

Taulukosta havaitaan että MC minimointi oli näissä systeemeissä huomattavasti nopeampi minimointimenetelmä. Toisaalta SD-menetelmä onnistui löytämään energiaminimin paremmin.

Kuvissa 1 ja 2 on esitetty konvergoituneet elektronitiheydet eri systeemeille joko MC- tai SD-minimoitien tuloksena. Havaitaan että ne ovat kvalitatiivisesti samankaltaisia, mikä olikin tämän työn tavoitteena. Lisäksi elektronitiheys on keskittynyt ytimien läheisyyteen, joten voidaan päätellä että tulokset ovat fysikaalisesti mielekkäitä.

Kuvissa 3 on esitetty energian konvergoituminen minimointiaskeleiden funktiona. Havaitaan että MC minimointi tuottaa liian korkean kokonaisenergian. Tämä saattaa johtua siitä että energia katsottiin konvergoituneeksi jos energian muutos oli pienempää kuin 10^{-4} H. Varmempi tapa olisi ollut vaatia että energian muutos olisi ollut pienempää kuin 10^{-4} H esimerkiksi viiden minimointiaskeleiden ajan. Toinen mahdollinen syy MC-minimoinnin huonompaan suoritukseen voi olla MC-minimoinneissa käytetty lämpötila. Tässä työssä käytettiin lämpötilaa 100 K. Korkeampaa lämpötilaa käyttämällä olisi todennäköisesti paremmin voitu välttää paikalliset energiaminimit.



Kuva 3: Kokonaisenergian konvergoiminen iteraatioaskelten funktiona. Input file oli 'alkuarvot.txt_10x10x0'. 'SD' tarkoittaa että minimointi tehtiin Steepest descent menetelmällä ja 'MC' että minimointiin Monte Carlo algoritmilla.

3 Työn mahdolliset puutteet ja parannusehdotukset

Ensinnä täytyy sanoa että muutamassa viikossa ei saada aikaakn täydellistä kvanttikemian ohjelmaa. Tässä työssä saatiin aikaan sopiva lähtökohta myöhemmälle mahdolliselle kehitystyölle.

Energian minimointia tulisi parantaa jotta globaali minimointi löydetään mahdollisimman hyvin. Tämä onnistuu tehokkaimin ja helpoimmin käyttämällä Scipy:n valmiita minimointirutiineita[Jones et al. (2001)]. Nyt etenkin rajoitettu optimointi (elektronimäärän piti säilyä vakiona) aiheutti ongelmia minimoinneissa.

Toinen puute ohjelman tämänhetikisessä versiossa on se että kaikki elektronit optimoidaan samanaikaisesti. Muutama vuosi sitten on saatu selville että tulokset orbitaalivapaassa tiheysfunktionaaliteoriassa paranevat huomattavasti jos eri kulumaliikemäärän omaavat elektronit optimoidaan erikseen[Ke et al. (2013)].

4 Javadocia vastaava dokumentaatio

Javadocia vastaava dokumentaatio on toteutettu Sphinx:illä. Se käynnistyy selaimella esim. komennolla 'firefox TiraLabra/Docs/html/index.html'

Viitteet

- Jones, E., Oliphant, T., and Peterson, P. (2001). Scipy: Open source scientific tools for python. *<http://www.scipy.org/>*.
- Ke, Y., Libisch, F., Xia, J., Wang, L.-W., and Carter, E. A. (2013). Angular-momentum-dependent orbital-free density functional theory. *Physical review letters*, 111(6):066402.