

Dijoditiofeeni-ionin dissosiaation mallinnus
klassisella molekyyldynamiikalla

Pro Gradu
Turun yliopisto
Fysiikka
2021
Kasper Juntunen
Tarkastajat:
Edwin Kukk
Johannes Niskanen

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck-järjestelmällä

TURUN YLIOPISTO

Fysiikan laitos

Juntunen Kasper Dijoditiofeeni-ionin dissosiaation mallinnus klassisella molekyyliidynamiikalla

Pro Gradu, 54 s.

Fysiikka

Elokuu 2021

Materiaalifysikaaliseen tutkimukseen liittyy usein fotoionisaation aiheuttama molekyylien hajoaminen, mikä antaa hyödyllistä tietoa muunmuassa molekyylien ominaisuuksista. Erilaisten tietokonesimulaatioiden käyttö samojen tapahtumien mallinnuksessa voi olla suureksi avuksi osana tutkimustyötä, mutta tätä vaikeuttaa itse fysikaalisen prosessin monimutkaisuus.

Molekyyliidynamiikka pitää sisällään monimutkaisuudeltaan erittäin vaihtelevia simulaatiometodeja, joista monet ovat laskennallisesti raskaita. Tässä tutkielmassa rakennetaan yksinkertainen klassiseen mekaniikkaan perustuva malli, jonka keveys mahdollistaa suuren määrän simulaation toistoja ja siten tulosten jakauman analyysin.

Molekyyliimallia käytetään ionisoidun dijoditiofeeni-molekyylin ($C_4H_2I_2S$) hajoamisen tutkimiseen antamalla sille alkuvaraus ja integroimalla systeemiä ajassa käyttäen Leapfrog-algoritmia. Varaukset liikkuvat simulaation aikana hyppimällä atomoilta toiselle. Simuloituja liikeratoja kerätään viisinumeroinen lukumäärä.

Mallin toimivuutta arvioidaan lopuksi vertaamalla saatuja tuloksia oikeiden molekyylien hajoamisessa syntyneisiin vastaaviin. Tuloksien ei havaita vastaavan kovin hyvin vertailudataa, mikä ei ole suuri yllätys. Potentiaalia on kuitenkin havaittavissa ja parametrien säätämällä sekä jatkokehityksellä, johon tutkielmassa esitetään ideoita, on mahdollista, että mallista saadaan käyttökelpoinen apu molekyylien hajoamiseen keskittävälle tutkijalle.

Asiasanat: Molekyyliidynamiikka, mallinnus, klassinen mekaniikka, simulaatio, molekyylin hajoaminen, varausten liikkuminen, Leapfrog

Sisällys

Johdanto	1
1 Molekyylin hajoamisen simulointi	2
1.1 Systemin esittely	3
1.2 Varausten liikkuminen	6
1.3 Integrointiprosessi	7
1.3.1 Atomien välillä vaikuttavat voimat	8
1.4 Parametrien valinta	11
2 Simulaatioiden tulosten käsittely ja esitys	14
2.1 Molekyylifragmenttien jakaumat	15
2.2 Sisäenergian jakaumat simulaatioissa	19
2.3 Hyppytaajuuksien vertailu	20
2.4 Molekyylifragmenttien tuoton riippuvuus sisäenergiasta	25
2.4.1 Alkuvarauksen +4 systeemin fragmenttien tuotot	25
2.4.2 Alkuvarauksen +2 systeemin fragmenttien tuotot	28
3 Tulosten vertailu	31
3.1 Fragmenttien jakauma	31
3.2 Fragmenttien tuotot energian funktiona	34
4 Yhteenveto	39
5 Liite 1: Lisää ionituottokuvaajia	43
6 Liite 2: Simulaation koodi	47

Johdanto

Materiaalifysikaalisessa tutkimuksessa molekyyliä ionisoidaan usein käyttäen esimerkiksi synktroneja tai hiukkaskiihdyttimiä, mikä voi johtaa hajoamistapahtumiin, missä varatut molekyylifragmentit saavat suuria kineettisiä energioita. Mittaussarjan tuottaman spektrin analyysi (esimerkiksi hajoamistapauksissa syntyneet fragmentit) on myös olennainen osa tällaista tutkimusta. Tutkijan työtä voikin helpottaa huomattavasti mahdollisten molekyylien hajoamisen simuloiminen joko etukäteen (kiinnostavien molekyylien valitsemiseen oikeita mittauksia varten) tai esimerkiksi mittausten jälkeisen analyysin tukena tulosten arvioimisessa.

Molekyyliidynamiikka, johon molekyylien hajoamisen simuloiminenkin kuuluu, on kuitenkin laaja ala täynnä eriasteisen monimutkaisuuden metodeja ongelman ratkaisuun. Yleisesti atomien ja niiden muodostamien molekyylien liikettä mallinetaan usein klassisen mekaniikan pohjalta laskien voimia potentiaaleista, mutta myös kvanttimekaniikkaa voidaan yhdistää siihen [1]. Ala on aktiivinen tutkimuksen kohde, ja sille on esitetty hyvin monipuolisia sovelluksia kuten esimerkiksi vesimolekyylien järjestäytymisen TiO_2 -pinnoille simuloiminen [2] ja lääkeaineen imeytymisen ennustaminen [3]. Usein molekyyliidynamiikan simulaatioita tehdään käyttäen pitkälle kehitettyjä ohjelmistoja, jollainen esimerkiksi simulaation aikaskaalan kiihdyttämisen mahdollistava LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) on [4].

Molekyylien simuloimisessa olennaista on myös elektronien käyttäytyminen. Elektronirakenteiden approksimaatioon on suuri määrä kvanttimekaniikkaan perustuvia metodeja, kuten Hartree-Fock -metodi [5] ja DFT (Density Functional Theory) [6]. Tämän tutkielman aihetta lähemmin sivuavasti myös varausten liikkumista molekyyliissä on mallinnettu käyttäen aikariippuvaa DFT:tä muunmuassa tutkimuksessa, jossa Folorunso *et al.* tutkivat varausten liikettä hiilivetyjaketjuissa [7].

Molekyyliidynamiikkaa ja *Ab initio* -menetelmiä on myös alettu yhdistää (elekt-

ronirakenteita lasketaan uudelleen simulaation edetessä), mikä mahdollistaa esimerkiksi kemiallisten sidosten hajoamisen ja muodostumisen mallintamisen [8].

Ongelma juuri molekyylien hajoamisen simuloimisessa on itse fysikaalisen tapahtuman monimutkaisuus, mistä johtuen siihen ei ole olemassa valmista ohjelmistoa. Kvanttimekaniikkaa käyttäen on mahdollista tehdä fysikaalisesti melko tarkka malli, mutta sekin olisi vain yksinkertaistus ja silti aivan liian raskas laskennallisesti. Käyttökelpoisen molekyyliidynaamisen mallin luominen vaatii sitä, että ongelmaa yksinkertaistetaan rajusti, ainakin jos mallin halutaan olevan helppo ja nopea apu tutkimuksen osana.

Tämän tutkielman tavoitteena onkin rakentaa malli, joka on yksinkertainen ja laskennallisesti kevyt, ja jolla voidaan tuottaa iso lukumäärä hajoamistapahtumia fragmenttien jakauman analysointia varten. Molekyyliidynamiikka oletetaan kokonaisuudessaan toimivaksi klassisen mekaniikan mukaisesti, eikä kvanttimekaanisia ilmiöitä huomioida mitenkään. Mallissa molekyylin ja sen dissosiaatioprosessin monimutkaiset fysikaaliset ominaisuudet pyritään huomioimaan käyttäen yksinkertaisia parametrejä, joita säätämällä olisi mahdollista päästä käsiksi hyödyllisiin lopputuloksiin.

1 Molekyylin hajoamisen simulointi

Tutkielman tavoitteena on edellisessä kappaleessa selitetysti rakentaa yksinkertainen malli molekyylin hajoamisen simuloimista varten. Oikeissa tutkimuksissa fotionisaatiolla molekyyliä voidaan ionisoida eri tavoilla riippuen laitteistosta, mikä vaikuttaa molekyylin saaman varauksen suuruuteen. Tässä mallissa keskitytään kahteen tavanomaiseen tapaukseen: tavallisen auger-ilmiön luoma +2-varaustila ja moniosaisen auger-ilmiön (cascade auger) luoma +4-tila.

Tässä kappaleessa käydään ensin lyhyesti läpi mallin pääpiirteet, minkä jälkeen tarkemmat yksityiskohdat simuloitavasta systeemistä, integraatioprosessista ja ato-

meihin vaikuttavista voimista, ja parametrisaatiosta on esitelty omissa alakappaleissaan. Koodi, jolla simulaatiot on tehty, löytyy liitteenä tutkielman lopusta.

Simulaatiossa tutkittavan molekyylin atomit ovat aloitettaessa sidostensa lepotilassa. Ainoastaan sidokselliset atomit ovat kytkettyjä toisiinsa Morse-potentiaalilla [9]. Atomien väliset sidokset eivät voi hajota, eivätkä alussa sidoksettomat atomit voi myöskään muodostaa uusia sidoksia simulaation aikana.

Systeemin varaustilalle on yllä selitetystä syystä kaksi vaihtoehtoa, +2 ja +4. Atomit, joille varaukset alussa annetaan, on valittu satunnaisesti valenssielektronien lukumäärän mukaisella painotuksella vastaamaan systeemin tilaa, missä Auger-ilmiössä syntyneet varaukset ovat jakaantuneet systeemiin.

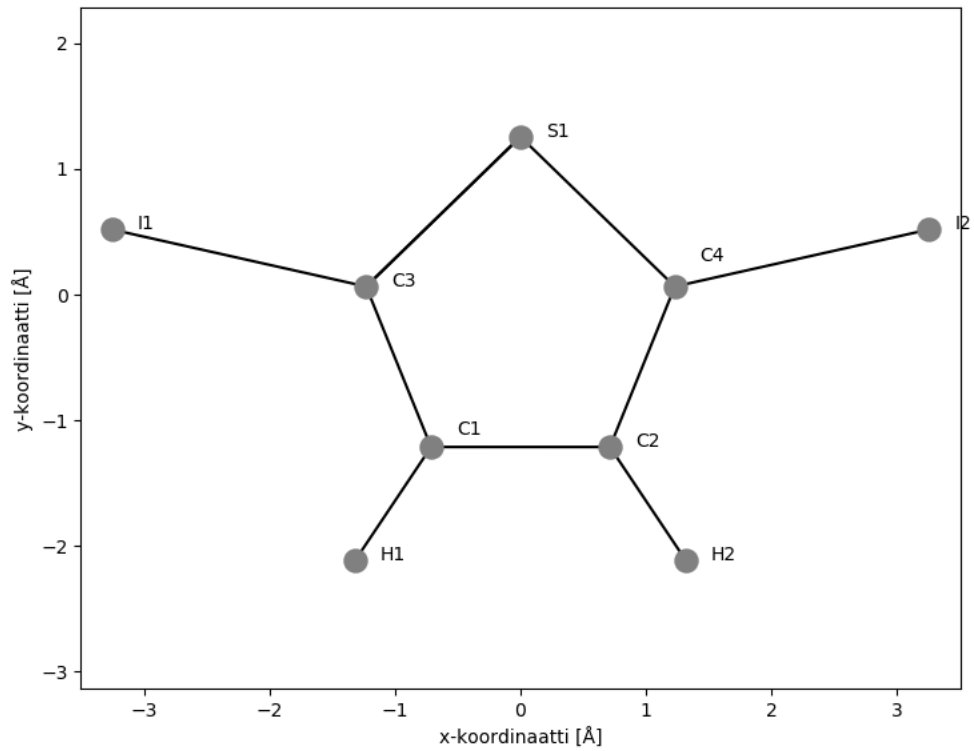
Atomien liikettä simuloidaan integroimalla Newtonin liikeyhtälöitä *aika-askel* -parametrin välein. Varausten liikkuvat atomien välillä simulaation aikana atomien välisen etäisyyden ollessa tarpeeksi pieni.

1.1 Systeemin esittely

Tässä tutkielmassa simulaation kohteena on molekyyli 2,5-dijoditiofeeni ($C_4H_2I_2S$, 2,5-Di-iodothiophene). Molekyyli on planaarinen, mistä johtuen kaikki dynamiikka tapahtuu kahdessa ulottuvuudessa. Tässä kappaleessa on esitelty kaikki simulaatiossa käytetyt arvot parametreille liittyen kyseiseen molekyyliin.

Kuvassa 1 on esitetty kuva molekyylin geometriasta, ja taulukossa Ia on esitetty simuloitavan molekyylin atomien alkupaikkojen koordinaatit [10], atomien massat [11], valenssielektronien lukumäärät [12] ja atomien minimisäteet hylkivää jousivoimaa varten. Taulukossa Ib on esitetty simulaation kulkuun, molekyylin sidoksiin ja varausten liikkumiseen liittyvät parametrit. Molekyylin sidosten energiat on kerätty keskiarvoina lähteistä [13] ja [14] ja sidosten vibraatioenergiakvanttien arvot on saatu sidosten värähtelytaajuuksista käyttäen näille tyypillisiä arvoja vaihteluvälin sisältä lähteestä [15].

Huomattavaa sidosten parametreissä on se, että sidosten vibraatioenergiakvanttien arvot vaikuttavat mallin toimintaan vähemmän kuin potentiaaliakuopan syvyydet, sillä ne muuttavat ainoastaan kuopan muotoa dissosiaatioenergian sijaan.



Kuva 1: Simuloidun diioditiofeenin geometria.

Taulukko I: Simuloidun molekyylin vakiot sekä systeemin parametrit.

(a) Atomien koordinaatit [10], atomimassat [11], valenssielektronien lukumäärät [12] ja minimisädeparametrit.

Atomi	x [Å]	y [Å]	m [u]	n_{el}	r_{min} [Å]
I1	-3,2564	0,519	126,90	7	1,038
I2	3,2563	0,5191	126,90	7	1,038
S1	-0,0001	1,2564	32,06	5	0,856
C1	-0,7124	-1,2119	12,011	4	0,542
C2	0,7125	-1,212	12,011	4	0,542
C3	-1,2297	0,06467	12,011	4	0,689
C4	1,2298	0,06467	12,011	4	0,689
H1	-1,3153	-2,1116	1,008	1	0,542
H2	1,3153	-2,1118	1,008	1	0,542

(b) Molekyylin sidosten energiat ([13] ja [14]), sidosten vibraatioenergiat [15] systeemin jousivakio sekä simulaation sisäiset hyppytaajuus, hyppyetäisyys, aika-askel, aika-askelten lukumäärä ja sisäinen energia.

Tunnus	Parametri	Arvo	Yksikkö
k	Jousivakio	1000	N/m
$D_{e,IC}$	I-C -sidosenergia	213	kJ/mol
$D_{e,SC}$	S-C -sidosenergia	272	kJ/mol
$D_{e,HC}$	H-C -sidosenergia	411	kJ/mol
$D_{e,CC}$	C-C -sidosenergia	346	kJ/mol
$E_{q,IC}$	I-C -vibraatioenergia	0,074	eV
$E_{q,SC}$	S-C -vibraatioenergia	0,086	eV
$E_{q,HC}$	H-C -vibraatioenergia	0,372	eV
$E_{q,CC}$	C-C -vibraatioenergia	0,185	eV
$f_{jump,1}$	Hyppytaajuus 1	$1 \cdot 10^{15}$	1/s
$f_{jump,2}$	Hyppytaajuus 2	$3,3 \cdot 10^{15}$	1/s
$r_{jump,max}$	Suurin hyppyetäisyys	2,5	Å
dt	Aika-askel	$0,5 \cdot 10^{-17}$	-
	Askelten lkm.	10^6	-
$E_{sis,2}$	Lisätty sisäenergia +2 systeemille	10	eV
$E_{sis,4}$	Lisätty sisäenergia +4 systeemille	0	eV

1.2 Varausten liikkuminen

Oikeissa molekyyleissä elektroneilla ei ole sijainteja, vaan niiden olinpaikat ovat ainoastaan todennäköisyyksiä aaltoluonteesta johtuen. Erilaisten molekyylien elektronien aaltofunktioita on mahdollista approksimoida numeerisesti, esimerkiksi käyttäen Hartree-Fock -menetelmää [5]. Tämä on kuitenkin laskennallisesti liian raskasta toteuttaa osana nopeaa simulaatiota, eikä muutenkaan sovellu osaksi haluttua yksinkertaista mallia. Tästä johtuen tutkielman mallissa varauksien todennäköisyyksiä olla eri paikoissa kuvataan yksinkertaisesti liikuttamalla kokonaisvarauksia molekyylin tai sen kappaleen sisällä huomioimatta sen vaikutusta potentiaalienergiaan.

Varausten liikkumisen nopeutta molekyyleissä on tutkittu käyttäen kvanttimekaniikkaa, jolla sille on saatu arvoja suuruusluokasta 2-5 Å/fs riippuen molekyylistä [7]. Tämän tutkielman mallissa siirtymät tapahtuvat äkillisesti, mutta todellista vastaavaan liikkumisnopeuden suuruusluokkaan päästään rajoittamalla siirtymien tapahtumistaajuus tiettyyn integraatioalgoritmin askelten lukumäärään (siirtymät tapahtuvat noin 1 fs välein).

Varausten liike on toteutettu teknisesti niin, että niillä integraatioalgoritmin askeleilla, joilla siirtymä sallitaan (parametrin *hyppytaajuus* rajoittama), jokainen atomi käydään yksitellen läpi. Kierron aikana niille atomeille, joilla on varaus, etsitään kaikki mahdolliset siirtymäkohteet, jotka täyttävät alla esitetyt kaksi sääntöä.

1. Siirtymät ovat mahdollisia vain sellaisten atomien välillä, jotka sijaitsevat enintään valitun maksimihyppyetäisyyden päässä (parametri $r_{jump,max}$) toisistaan siirtymähetkellä.
2. Varausten on mahdollista siirtyä ainoastaan sellaiseen atomiin, jonka varaus alkuperäisen atomin varausta on pienempi (esimerkiksi +1 varauksellisesta atomista voi siirtyä yksi varaus neutraalille atomille, mutta ei toiselle +1 varaukselliselle atomille).

Siirtymäkohteet painotetaan vastaavien atomien valenssielektronien lukumäärällä, josta on poistettu atomien mahdollinen varaus. Lopuksi uusi sijainti valitaan satumanvaraisesti painotuksen mukaisesti.

Yksinkertaistettuna esimerkkinä kolmen atomin molekyyllisessä siirtymä +2 varauksellisesta hiiliatomista neutraaliin jodi-atomiin (7 valenssielektronia) on suhteessa 7:3 todennäköisempää kuin siirtymä +1 varaukselliseen hiiliatomiin (3 valenssielektronia ($4 - 1 = 3$)).

1.3 Integrointiprosessi

Simulaatiossa atomien liikettä integroidaan numeerisesti käyttäen Leapfrog-algoritmia, jossa atomien nopeuksia lasketaan puolikkaan integraatioaskeleen verran atomien paikkoja edellä [16]. Algoritmi on valittu siksi, että se on helppo toteuttaa ja on laskennallisesti kevyt.

Simulaatio alkaa tilanteesta, jossa molekyylin atomit ovat paikallaan ja loppuu, kun parametrin *Askelten lukumäärä* asettama määrä integraatiokierroksia on kulunut.

Leapfrog-algoritmin toiminta on esitetty lyhyesti alla.

1. Lasketaan atomien kiihtyvyydet, jotka riippuvat vain paikasta

$$\vec{a}(t) = \frac{F(\vec{r}(t))}{m}$$

2. Lasketaan atomeille puoliaskelnopeudet kiihtyvyyksien avulla

$$\vec{v}(t + \frac{1}{2}\Delta t) = \vec{v}(t) + \vec{a}(t)\frac{1}{2}\Delta t.$$

3. Lasketaan atomien uudet paikat käyttäen puoliaskelnopeutta

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t + \frac{1}{2}\Delta t)\Delta t$$

1.3.1 Atomien välillä vaikuttavat voimat

Integrointiprosessin vaiheessa 3 uudet kiihtyvyydet atomeille lasketaan Newtonin toisen lain mukaan käyttäen atomeihin kohdistuvia voimia. Tässä simulaatiossa kahden atomin välillä on mahdollista olla kolmen eri tyyppin voimia, jotka on esitetty yhtälöissä 4, 7 ja 8. Ensimmäinen näistä voimista vaikuttaa ainoastaan sidoksissa olevien atomien välillä ja on peräisin sidoksia kuvaavasta Morse-potentiaalista [9].

$$E_{\text{morse}} = D_e \cdot (1 - e^{-b \cdot (r - R_e)})^2 - D_e, \quad (1)$$

jossa D_e on sidoksen potentiaaliukuopan syvyys, R_e atomien välisen sidoksen pituus, r atomien euklidinen etäisyys toisistaan ja b on määritelty alla esitetyllä tavalla, jossa E_q on sidoksen värähtelyn vibraatioenergiakvantti, \hbar on redusoitu Planckin vakio ja m on redusoitu massa.

$$b = \frac{\frac{E_q}{2}}{\hbar \sqrt{\frac{2m}{D_e}}} \quad (2)$$

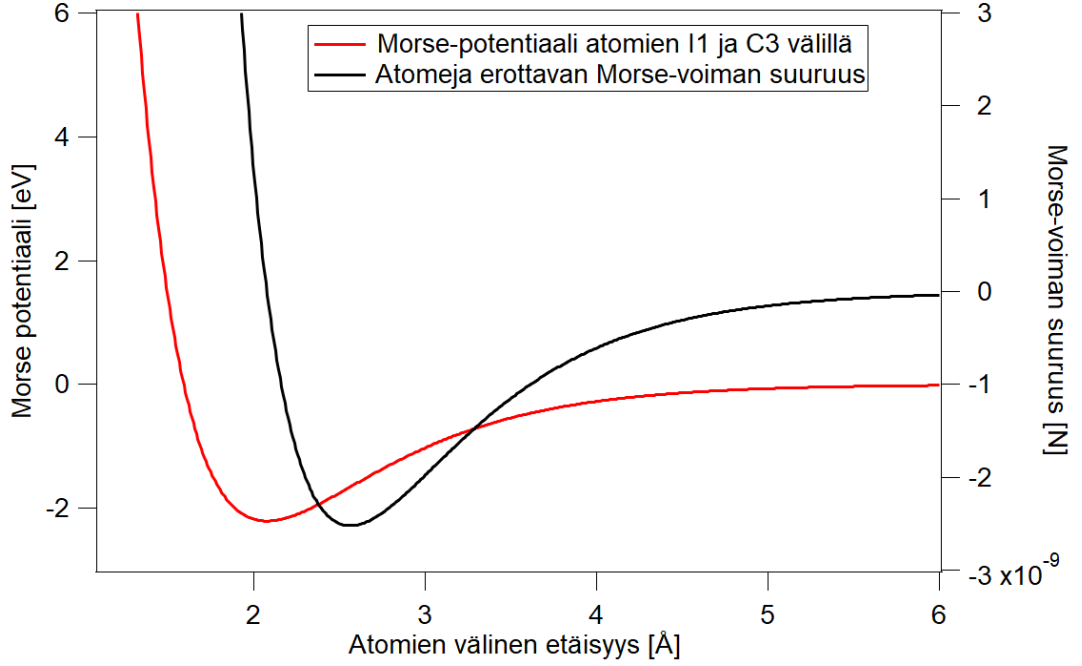
Redusoitu massa on puolestaan laskettu sidoksen muodostavien atomien (i ja j) massoista (esitetty taulukossa Ia) alla esitetyllä tavalla.

$$m = \frac{1}{\frac{1}{m_i} + \frac{1}{m_j}} \quad (3)$$

Morse-potentiaalin derivaattana saadaan siitä aiheutuva voima, joka vaikuttaa atomien i ja j välillä. Atomiin i kohdistuu voima, joka on alla esitettyä muotoa, ja atomiin j kohdistuu yhtä suuri vastakkaiseen suuntaan osoittava voima.

$$\begin{aligned} \vec{F}_{\text{morse},ij} &= 2bD_{e,ij} e^{b(R_{e,ij} - r_{ij})} (1 - e^{b(R_{e,ij} - r_{ij})}) \hat{e}_{ij} \\ \vec{F}_{\text{morse},ji} &= -\vec{F}_{\text{morse},ij} \end{aligned} \quad (4)$$

Yhtälössä \hat{e}_{ij} on yksikkövektori atomista i atomia j kohti. Kuvassa 2 on esitetty Morse-potentiaali sekä siitä aiheutuva voima atomeja erotavaan suuntaan (paikat molekyyllissä kuvassa 1) atomien välisen etäisyyden funktiona.



Kuva 2: Atomien I1 ja C3 välinen Morse-potentiaali ja atomeihin kohdistuvan voiman suuruus atomeja erottavaan suuntaan atomien välisen etäisyyden funktiona.

Atomien välisten sidosten vahvuuksien oletetaan myös riippuvan atomien varauksesta, sillä valenssielektronit, joiden puuttuminen luo atomille varauksen, osallistuvat sidosten muodostamiseen. Tutkielman mallin yksinkertaisesta luonteesta johtuen varausten oletetaan heikentävän sidoksia lineaarisesti. Tällöin, jos atomit ovat varattuja, alkuperäinen potentiaali kuopan syvyys pienenee riippuen kummankin atomin valenssielektronien lukumäärästä n_{el} ja varauksesta q

$$D_{e,varattu} = D_e \frac{(n_{el,i} - q_i) + (n_{el,j} - q_j)}{n_{el,i} + n_{el,j}}, \quad (5)$$

jossa varaukset ovat kokonaislukuina vastaten yksittäisten elektronien puuttumista.

Simulaation parametri *lisätty sisäenergia* vaikuttaa myös molekyylin sidoksiin. Mallissa oletetaan naiivisti, että kaikki systeemiin lisätty energia jakautuu tasaisesti sidoksille ja heikentää niitä. Potentiaali kuopan syvyyttä pienentää tällöin lisätty sisäenergia (kokonaismäärä E_{sis} jaettuna sidosten lukumäärällä n_{sidos}) alla esitetyllä

tavalla.

$$D_{e,\text{todellinen}} = D_{e,\text{varattu}} - \frac{E_{\text{sis}}}{n_{\text{sidos}}} \quad (6)$$

Sidoksellisiin atomeihin vaikuttavien voimien lisäksi jokaisen atomin välillä vaikuttaa myös Coulombinen repulsio, jonka suuruus saadaan Coulombin potentiaalista [17]. Coulombin voima atomien i ja j välillä on alla esitettyä muotoa, jossa voima vaikuttaa atomiin i ja kohti atomia j . Atomiin j vaikuttaa yhtä suuri voima vastakkaiseen suuntaan.

$$\begin{aligned} \vec{F}_{\text{coulomb},ij} &= -k_e \frac{q_i \cdot q_j}{r_{ij}^2} \hat{e}_{ij} \\ \vec{F}_{\text{coulomb},ji} &= -\vec{F}_{\text{coulomb},ij}, \end{aligned} \quad (7)$$

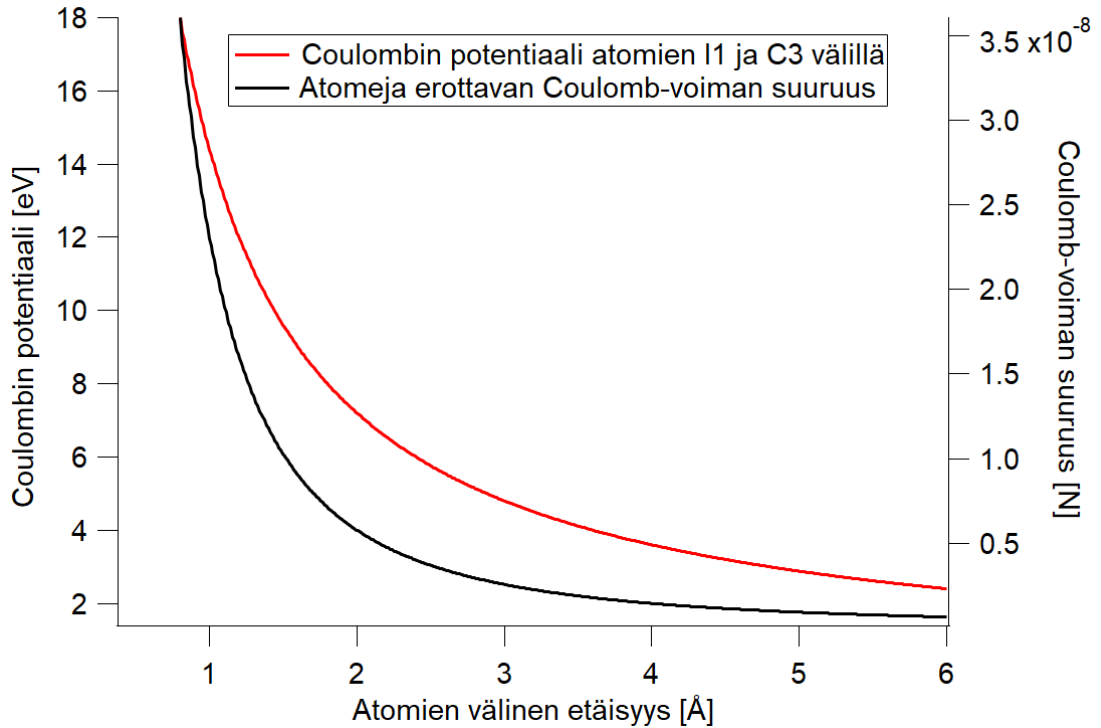
jossa k_e on Coulombin vakio, q_i ja q_j ovat atomien i ja j varaukset ja r_{ij} atomien välinen etäisyys. Atomien I1 ja C3 (paikat molekyyllissä kuvassa 1) välinen Coulombin potentiaali ja Coulomb-voiman suuruus on esitetty kuvassa 3 molempien atomien varauksen ollessa $+1$.

Viimeinen kolmesta voimasta on jokaisen mallissa sidoksettoman atomiparin välillä vallitseva Hooken lain mukainen jousivoima, joka estää atomeja päätyvästä liian lähelle toisiaan. Käytännössä tämä on toteutettu niin, että kahden atomin välisen etäisyyden ollessa pienempi kuin atomien minimisädeparametrien summa, atomiin i kohdistuu alla esitetyn mukainen hylkivä voima atomista j ja atomiin j kohdistuu luonnollisesti yhtä suuri vastakkaisuuntainen voima.

$$\begin{aligned} \vec{F}_{\text{jousi},ij} &= -k \cdot ((r_{\text{min},i} + r_{\text{min},j}) - r_{ij}) \hat{e}_{ij} \\ \vec{F}_{\text{jousi},ji} &= -\vec{F}_{\text{jousi},ij}, \end{aligned} \quad (8)$$

jossa k on systeemin jousivakio-parametri, $r_{\text{min},i}$ ja $r_{\text{min},j}$ ovat atomien minimisädeparametrit ja r_{ij} on atomien etäisyys toisistaan.

Kuvassa 4 on esitetty atomien I1 ($+2$ varaus) ja C3 ($+1$ varaus) (paikat molekyyllissä kuvassa 1) väliset voimat ja atomien välinen etäisyys simulaation aikana yksinkertaistetussa esimerkkisimulaatiossa, jossa varausten liike on kytketty pois päältä.

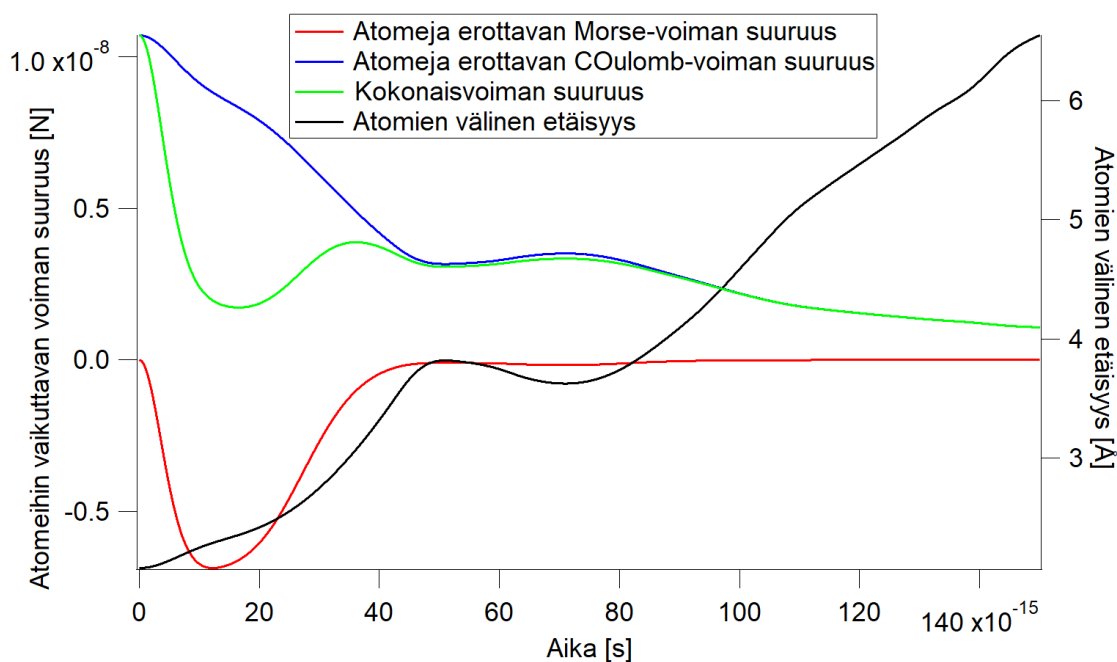


Kuva 3: Atomien I1 ja C3 välinen Coulombin potentiaali ja siitä aiheutuvan atomeja erottavan voiman suuruus atomien välisen etäisyyden funktiona.

1.4 Parametrien valinta

Simulaatiossa on useita eri parametrejä sekä atomeille, että molekyylille, jotka on valittu eri tavoin. Kappaleessa 1.1 on esitetty sekä kirjallisuuslähteistä kerätyt arvot fysikaalisille parametreille että tutkielman osana itse valitut lukuarvot muille parametreille. Jälkimmäisten arvojen valintaperusteita on selostettuna tässä kappaleessa.

Sidoksettomien atomiparien välinen hylkivä jousivoima ei ollut osa mallia alussa, mutta simulaatioiden aikana molekyyli käyttäytyi ilman sitä hyvin epäfysikaalisesti. Tämä ilmeni esimerkiksi niin, että keskenään sidoksettomat atomit saattoivat liikkua erittäin lähelle toisiaan, jolloin varausten hyppiessä saattoi syntyä tilanteita, missä nämä atomit saivat varauksen ja siten massiivisen (mallin normaaliin käyttäytymiseen verrattuna) kineettisen energian Coulomb-voimasta. Yksinkertaisin keino



Kuva 4: Atomien I1 ja C3 väliset etäisyys ja atomeihin vaikuttavien voimien suuruudet atomeja erottavaan suuntaan yksinkertaistetussa esimerkksimulaatiossa.

estää tämä oli asettaa atomeille säde, joiden summaa pienemmän etäisyyden sisällä atomeja työntää erilleen energian säilyttävä jousivoima.

Atomien välillä vallitsevan jousivoiman jousivakio k on valittu siten, että voima riittää pitämään atomit erillään, mutta ei aiheuta liian suuria numeerisista seikoista johtuvia muutoksia atomien liikeratoihin tai lennätä minimisäteen sisälle hetkellisesti päätyneitä atomeja irti molekyylistä. Testien perusteella tähän sopivaksi arvoksi valikoitui 1000 N/m, minkä ei havaittu vaikuttavan numeeriseen tarkkuuteen negatiivisesti.

Atomien minimisädeparametri (esitetty taulukossa Ib), johon jousivoima perustuu, on laskettu molekyylin alkugeometriasta. Jokaisen atomin minimisäde on puolikas lyhimmästä euklidisesta etäisyydestä, joka kyseisellä atomilla on toiseen atomiin alkugeometriassa. Esimerkkinä tästä jodi-atomi, jonka lähin naapuri on siinä sidoksissa oleva hiiliatomi (ks. molekyylin geometria kuvassa 1).

$$\begin{aligned}
r_{min,I} &= \frac{\sqrt{(x_C - x_I)^2 + (y_C - y_I)^2}}{2} \\
&= \frac{\sqrt{(-1,2297 \text{ \AA} + 3,2564 \text{ \AA})^2 + (0,06467 \text{ \AA} - 0,519 \text{ \AA})^2}}{2} \\
&= 1,038499955 \text{ \AA} \\
&\approx 1,0385 \text{ \AA}
\end{aligned} \tag{9}$$

Simulaation aika-askeleen valinnassa olennaista on integraatioprosessissa syntyvän virheen minimoiminen, joka syntyy siitä, että toisen asteen integraatioalgoritminä Leapfrog pitää sisällään $O((\Delta t)^3)$ -suuruisen virhetermin [16]. Pienimmän mahdollisimman askeleen käyttö parantaa simulaation tarkkuutta, mutta myös pidentää simulaation ajallista kestoja. Sopivaksi arvoksi parametrille valikoitui $0,5 \cdot 10^{-17}$ s, jolla yksittäinen simulaatio ei kuluta niin pitkää aikaa, että niiden toistaminen tuhansia kertoja kävisi mahdottomaksi käytössä olevilla laskentaresursseilla.

Simulaation integraation kokonaiskesto on valittu niin, että se on käytössä olevien laskentaresurssien kannalta suurin mahdollinen, jolla simulaation toistoja on mahdollista suorittaa tarpeeksi. Valittu aika-askel huomioiden tähän päästään miljoonalla integraatioaskeleella, jolloin kokonaissimulaatioaika on 5 ps ja laskennan ajallinen kesto 5-10 minuuttia.

Alkuvarauksen +2 systeemi on tutkielman simulaatioissa stabiili, eli se ei hajoa simulaatiossa ilman ylimääräistä energiaa. Kuitenkin esimerkiksi kappaleessa 3 esitetystä tutkimusdatasta voidaan huomata todellisen molekyylin hajoavan tällä varauksella, joten molekyylille on annettava simulaatiossa lisäenergiaa jossain muodossa. Malliin ei sen yksinkertaisuudesta johtuen kuulu lämpötila tai sen aiheuttama atomeille annettava satunnaisliike, vaan energian oletetaan yksinkertaisesti heikentävän sidoksia. Tässä simulatiossa se on toteutettu *lisätty sisäenergia* -parametrilla, jonka sisältämä energiamäärä jaetaan kaikkien molekyylin sidosten kesken tasan. Sopivaksi energiamääräksi, joka mahdollistaa molekyylin hajoamisen, valikoitui testien perusteella 10 eV.

Varausten hyppimistä atomista toiseen (selitetty tarkemmin kappaleessa 1.2) rajoittava parametri hyppimistaajuus on valittu kokeilemalla useita eri taajuuksia ja tarkastelemalla simulaation lopputuloksena syntyneitä hajooneen molekyylin kappaleita. Hyppimistaajuuden ollessa liian suuri suhteutettuna simulaation aika-askeleeseen (esimerkiksi hyppiminen tapahtuu joka toisella askeleella), varaukset liikkuvat liikaa eikä molekyyli pääse hajoamaan ollenkaan. Sama efekti on havaittavissa myös liian pienellä taajuudella, jolloin hajoaminen riippuu lähes täysin alkutilan varausten sijainneista. Parametrin arvo valittiin näiden kahden ääripään väliltä siten, että molekyyli hajosi käytännössä aina, mikä toteutui parhaiten arvolla $1 \cdot 10^{15}$ 1/s. Hyppimistaajuuden vaikutuksen arvioimista varten trajektoreja simuloitiin myös hyppytaajuudella $3,3 \cdot 10^{15}$ 1/s.

Toisen varausten hyppimiseen olennaisesti liittyvän parametrin, *suurin hyppyetäisyys* ($r_{\text{jump,max}}$), arvoksi valikoitui $2,5 \text{ \AA}$. Parametrin arvon vaikutusta ei ollut ajan puolesta mahdollista testata, joten valinta perustui yksinkertaisesti siihen, että arvo on tarpeeksi suuri mahdollistamaan siirtymät ainoastaan ehjän molekyylin tai sen fragmenttien sisällä.

2 Simulaatioiden tulosten käsittely ja esitys

Tutkielman tekemisen aikana ajettujen simulaatioiden kokonaismäärää rajoitti käytännössä ainoastaan käytössä olevat laskentaresurssit ja siten kulunut kokonaisaika. Lopulta simuloituja trajektoreja kertyi 20 000 kpl neljällä alkuvarauksella, 15 600 kpl kahdella alkuvarauksella käyttäen hyppytaajuutta $1 \cdot 10^{15}$ 1/s ja 5000 kpl käyttäen hyppytaajuutta $3,3 \cdot 10^{15}$ 1/s. Tässä kappaleessa on esitelty kaikki simulaatioissa syntynyt käsiteltävä data ja niistä johdetut tulokset.

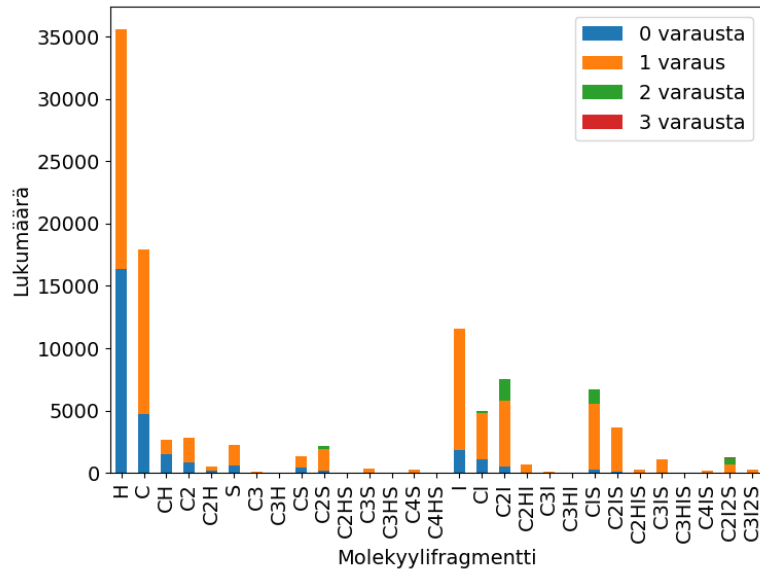
2.1 Molekyylifragmenttien jakaumat

Ensimmäinen tulos, joka simulaatioista saadaan, on hajonneiden molekyylien fragmenttien jakauma. Kuvassa 5a on esitetty alkuvarauksella +4 syntyneet molekyylifragmentit jaoteltuna lopullisen varauksen mukaan. Kuvassa 5b on esitetty sama alkuvarauksen +2 simulaatioille käyttäen hyppytaajuutta $1 \cdot 10^{15}$ 1/s. Alkuvarauksen +4 simulaatioissa lisätty sisäenergia on aina 0 eV ja alkuvarauksen +2 simulaatioissa aina 10 eV. Taulukossa II on esitetty fragmenttien lukumäärät numeroina.

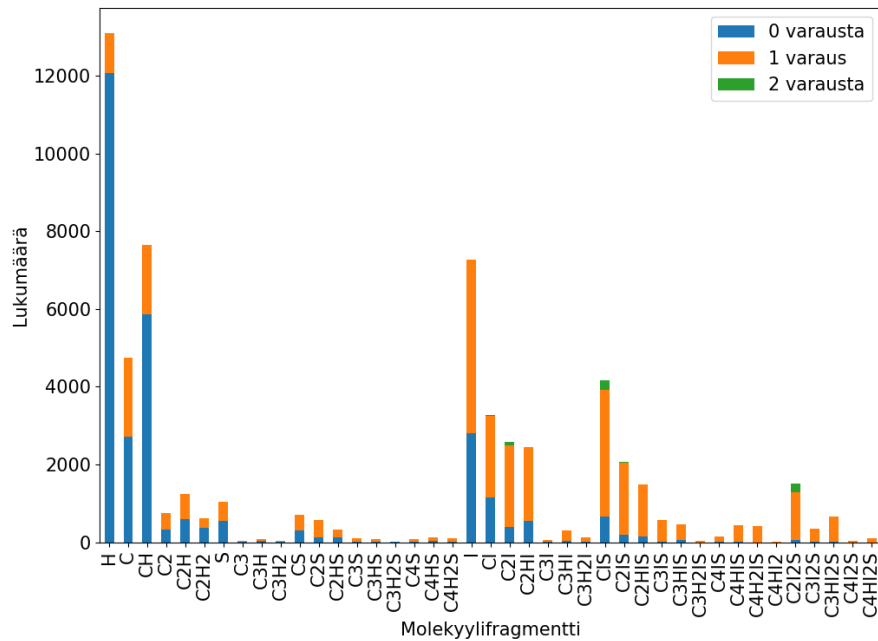
Neljän alkuvarauksen systeemillä molekyyli hajosi aina. Suurin kokonaisuena selvinnyt fragmentti oli C_3I_2S , eli alkuperäisestä molekyylistä oli irronnut aina vähintään yksi hiiliatomi ja molemmat vetyatomit. Jakaumasta voidaan myös nähdä, että vetyatomit irtoavat hyvin helposti isommista kappaleista ja usein jäävät varatuiksi.

Kahden alkuvarauksen systeemillä ($1 \cdot 10^{15}$ 1/s hyppytaajuus) isompia fragmentteja on jäänyt jäljelle pieni osa, mutta jokaisessa trajektorissa alkuperäisestä molekyylistä on irronnut vähintään yksi vetyatomi. Jakaumasta voidaan havaita neljän alkuvarauksen systeemin tapaan, että vetyatomeja irtoaa paljon, mutta mielenkiintoisesti suurin osa niistä on varauksettomia. Sama efekti voidaan havaita pienemässä määrin yksittäisillä hiiliatomeilla.

Valitun hyppytaajuusparametrin vaikutuksen arvioimiseksi +2 alkuvarauksen systeemiä simuloitiin vielä lopuksi käyttäen hyppytaajuutta $3,3 \cdot 10^{15}$ 1/s. Tästä syntyneet fragmentit on esitetty kuvassa 6. Käyttäen korkeampaa hyppytaajuutta suurimassaiset fragmentit jäivät useammin +2-varautuneiksi ja pieni osa molekyyleistä jäi myös kokonaan ehjiksi. Tarkempi vertailu näiden kahden sarjan välillä on tehty kappaleessa 2.3.

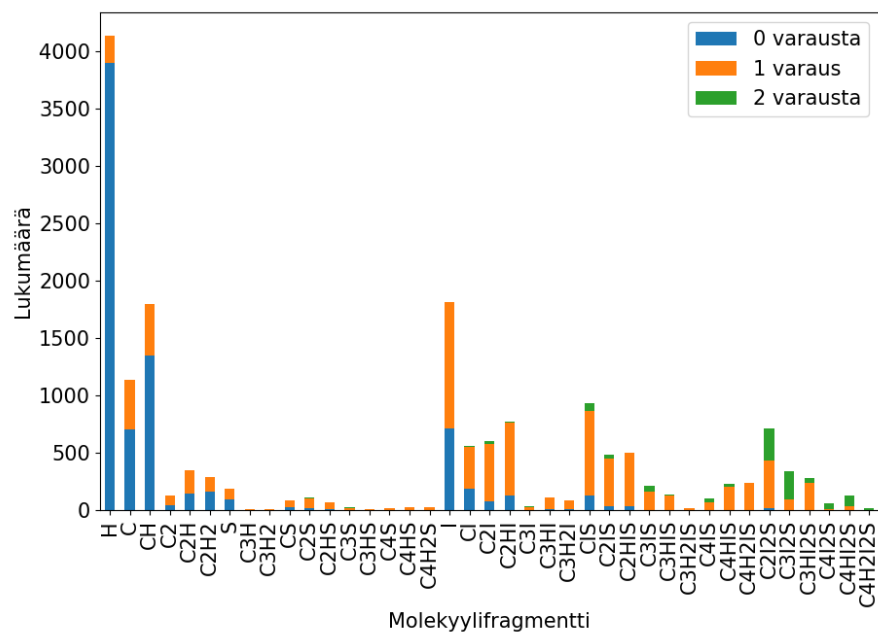


(a) Alkuvarauksen +4, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 0 eV simulaatioissa syntyneet fragmentit.



(b) Alkuvarauksen +2, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 10 eV simulaatioissa syntyneet fragmentit.

Kuva 5: Simulaatioissa syntyneiden molekyylifragmenttien jakaumat.



Taulukko II: Simulaatioissa syntyneiden fragmenttien lukumäärät jaoteltuna loppuvarauksen mukaan. Alkuvarauksen +2 simulaatioista on esitettyä ainoastaan hyp-pytaajuuden $1 \cdot 10^{15}$ 1/s data.

Alkuvarauksen +4 simulaatiot				
Fragmentti	0 varausta	1 varaus	2 varausta	3 varausta
H	16363	19263	0	0
C	4695	13246	0	0
CH	1493	1162	0	0
C ₂	813	2041	0	0
C ₂ H	156	375	0	0
S	629	1603	0	0
C ₃	29	69	0	0
C ₃ H	6	30	0	0
CS	460	914	6	0
C ₂ S	218	1676	282	0
C ₂ HS	15	44	0	0
C ₃ S	44	294	2	0
C ₃ HS	5	12	0	0
C ₄ S	23	254	0	0
C ₄ HS	0	25	0	0
I	1826	9779	0	0
CI	1075	3748	149	0
C ₂ I	545	5217	1768	0
C ₂ HI	44	668	0	0
C ₃ I	20	111	0	0
C ₃ HI	0	32	0	0
CIS	304	5250	1188	0
C ₂ IS	94	3519	51	3
C ₂ HIS	3	239	0	0
C ₃ IS	20	1041	20	2
C ₃ HIS	0	57	0	0
C ₄ IS	0	205	0	0
C ₂ I ₂ S	4	680	484	77
C ₃ I ₂ S	0	232	21	0

Alkuvarauksen +2 simulaatiot			
Fragmentti	0 varausta	1 varaus	2 varausta
H	12058	1038	0
C	2725	2012	0
CH	5871	1771	0
C ₂	330	417	0
C ₂ H	586	646	0
C ₂ H ₂	366	249	0
S	544	487	0
C ₃	12	18	0
C ₃ H	31	50	0
C ₃ H ₂	24	9	0
CS	305	400	0
C ₂ S	130	433	10
C ₂ HS	115	216	0
C ₃ S	17	84	0
C ₃ HS	20	54	0
C ₃ H ₂ S	5	10	0
C ₄ S	14	65	0
C ₄ HS	27	105	0
C ₄ H ₂ S	20	77	0
I	2797	4461	0
CI	1144	2098	29
C ₂ I	382	2110	77
C ₂ HI	538	1903	0
C ₃ I	8	58	0
C ₃ HI	32	266	0
C ₃ H ₂ I	22	99	0
CIS	666	3250	245
C ₂ IS	190	1853	15
C ₂ HIS	152	1344	0
C ₃ IS	20	553	9
C ₃ HIS	52	414	1
C ₃ H ₂ IS	0	39	0
C ₄ IS	8	144	0
C ₄ HIS	10	438	0
C ₄ H ₂ IS	0	413	0
C ₄ HI ₂	0	15	0
C ₂ I ₂ S	54	1223	224
C ₃ I ₂ S	3	340	8
C ₃ HI ₂ S	14	641	0
C ₄ I ₂ S	0	42	0
C ₄ HI ₂ S	0	100	0

2.2 Sisäenergian jakaumat simulaatioissa

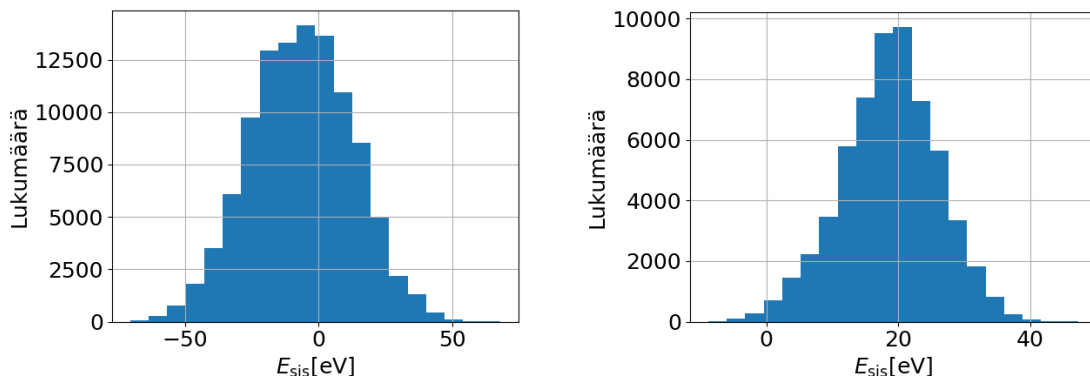
Simulaatioissa systeemin sisäinen energia ei pysy vakiona johtuen käytetyn varausten liikkumisen mallin yksinkertaisuudesta. Varausten hyppiminen ei ota huomioon aiheuttamaansa muutosta sähköiseen potentiaaliin, joten varausten liikkeessä satumanvaraisesti simulaation aikana energiaa syntyy ja häviää riippuen varausten sijainnista. Tämän efektin tarkastelua varten simulaatioista tallennettiin systeemin kokonaisenergian (kineettinen energia, sähköinen potentiaalienergia, sidosten potentiaalienergia) muutos alun ja lopun välillä. Tämän energianmuutoksen ja alussa systeemille annetun lisäenergian voidaan ajatella yhdessä muodostavan sisäisen energian E_{sis} .

$$E_{\text{sis,sim}}(t) = E_{\text{kin}}(t) + E_{\text{coul}}(t) + E_{\text{morse}}(t) - E_{\text{kin}}(0) - E_{\text{coul}}(0) - E_{\text{morse}}(0) + E_{\text{sis,lisatty}} \quad (10)$$

Todellisuudessa molekyylin sisäistä energiaa voi vapautua systeemin vaihtaessa elektronista tilaansa, minkä mallintaminen vaatisi elektronirakenteiden laskemisen ja lisäksi mekanismin elektronien siirtymille eri energiatasojen välillä, mikä ei kuulu tutkielmaan ongelman vaikeustason eikä tutkielman tavoitteidenkaan takia. Tätä prosessia kuvataan mallissa yksinkertaisesti yllä esitetyllä tavalla, jossa varausten hyppiminen muuttaa systeemin sisäenergian määrää.

Jakauma systeemin sisäisestä energiasta on esitetty kuvassa 7a alkuvarauksen +4 systeemille ja kuvassa 7b alkuvarauksen +2 systeemille. Jakaumien voidaan havaita olevan hyvin symmetrisiä ja melko leveitä. Systeemin energian lisääntyminen simulaation aikana on mahdollista perustella fysikaalisesti yllä esitetyllä tavalla, mutta sen vähenemiseen sama peruste ei käy, mistä johtuen tulosten tarkastelussa trajektorit, joissa sisäinen energia on vähentynyt, jätetään huomiotta.

Kuvissa 8a ja 8b ja taulukossa III on esitetty syntyneiden fragmenttien lukumäärät, kun kaikki negatiivisen energianmuutoksen trajektorit on hylätty. Alkuvarauksen +2 simulaatioissa poistettuja trajektoreja on erittäin vähän, koska systeemille



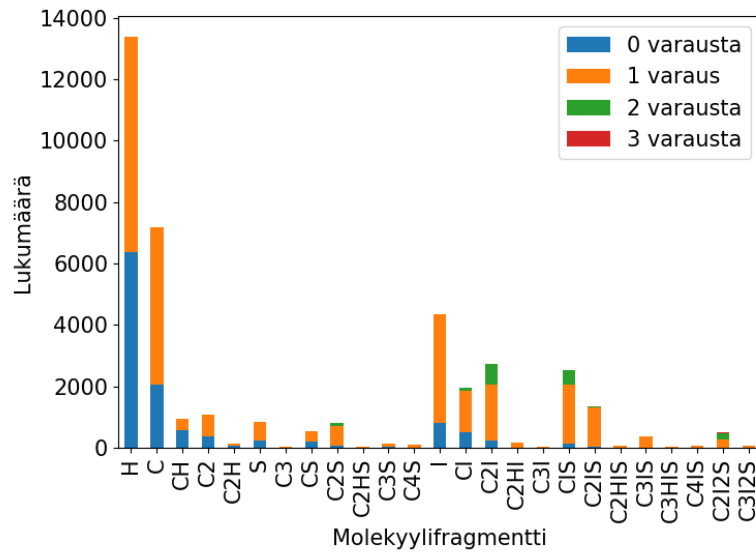
(a) Alkuvarauksen +4, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 0 eV simulaatioiden sisäisen energian jakauma. (b) Alkuvarauksen +2, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 10 eV simulaatioiden sisäisen energian jakauma.

Kuva 7: Simuloitujen trajektorien sisäenergiajakaumat.

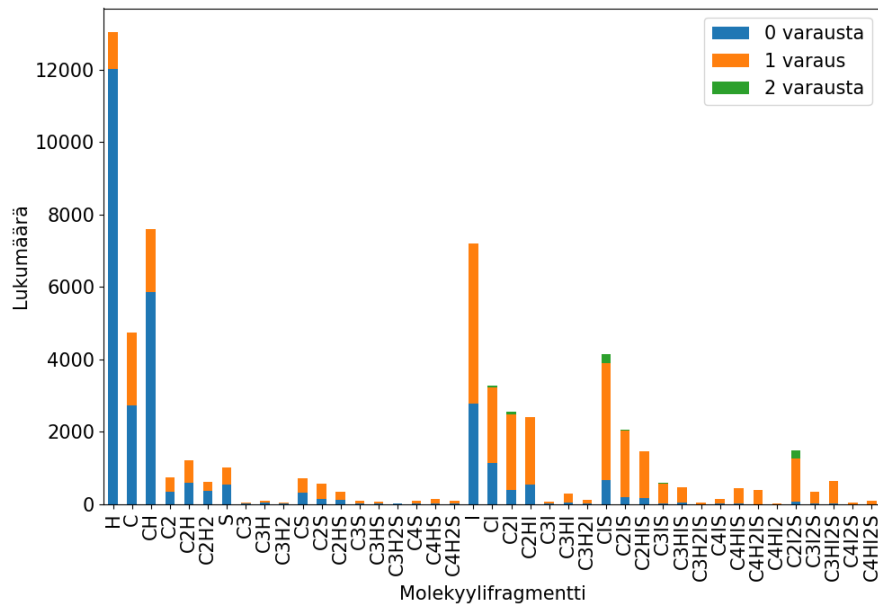
on annettu alussa ylimääräistä sisäenergiaa, joka on laskettu mukaan arvoihin, mikä siirtää jakaumaa positiiviseen suuntaan. Jakaumissa ei ole havaittavissa selviä eroja pienemmän trajektorilukumäärän lisäksi.

2.3 Hyppytaajuuksien vertailu

Eri hyppytaajuuksien aiheuttaman eron arvioimiseksi molemmilla saadut fragmenttijaumat on esitetty yhdessä normitettuna simuloitujen trajektorien määrän mukaan kuvissa 9a ja 9b. Fragmenttijauman esittelyn yhteydessä mainittu ero +2-varauksellisten fragmenttien suhteellisesta lukumäärässä on helppo havaita, minkä lisäksi suurimassaisia fragmentteja syntyi poikkeuksetta enemmän hyppytaajuuden $3,3 \cdot 10^{15}$ 1/s simulaatioissa. Vastaavasti pienemmän massan fragmentteja on luonnollisesti suhteellisesti enemmän hyppytaajuuden $1 \cdot 10^{15}$ 1/s simulaatioissa. Yksittäisten vetyjen määrissä ei ole havaittavissa suurta eroa ja pääosin fragmenttien varausten suhteet pysyvät melko samoina molemmilla taajuuksilla. Hyppytaajuusparametrillä voidaan päätellä olevan ainakin se vaikutus, että molekyyli hajoaa vä-



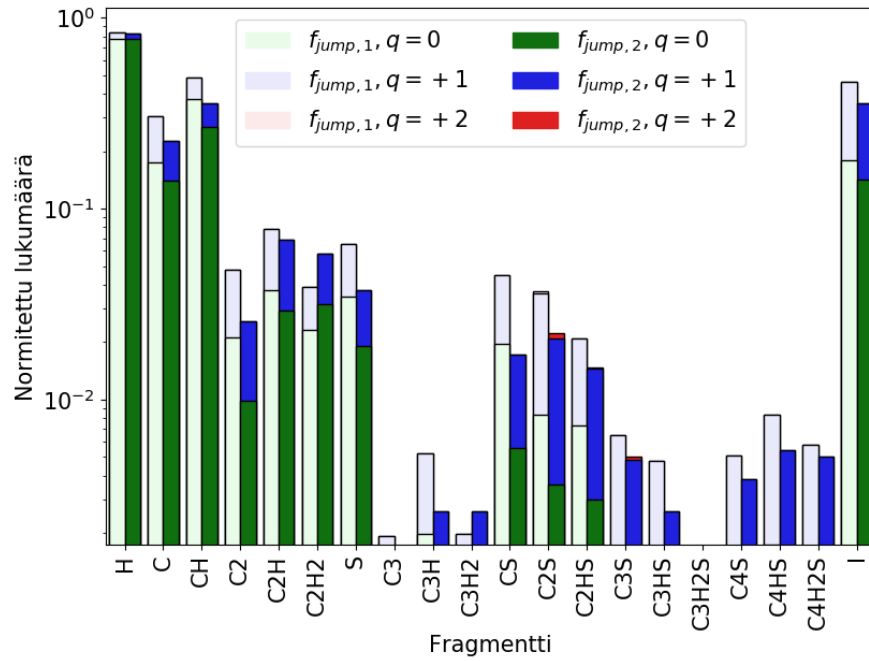
(a) Alkuvarauksen +4, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 0 eV simulaatioissa syntyneet fragmentit.



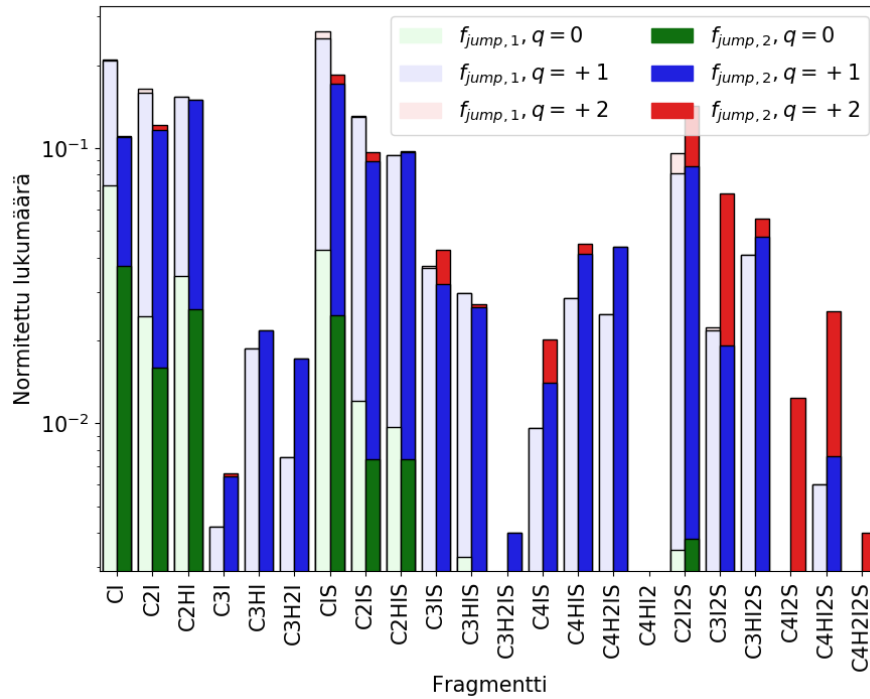
(b) Alkuvarauksen +2, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 10 eV simulaatioissa syntyneet fragmentit.

Kuva 8: Simulaatioissa syntyneiden molekyylifragmenttien jakaumat ilman negatiivisen sisäenergian trajektoreja.

hemmän varausten hyppiessä useammin.



(a) Pienimassaiset fragmentit.



(b) Suurimassaiset fragmentit.

Kuva 9: Alkuvarauksen +2 simulaatioiden tuottamien fragmenttien normitetut lukumäärät varauksen (q) mukaan lajiteltuna hyppytajuuksilla $f_{jump,1} = 1 \cdot 10^{15}$ 1/s ja $f_{jump,2} = 3,3 \cdot 10^{15}$ 1/s.

Taulukko III: Simulaatioissa syntyneiden fragmenttien lukumäärät ilman negatiivisen sisäenergian trajektoreja jaoteltuna loppuvarauksen mukaan. Alkuvarauksen +2 simulaatioista on esitettyä ainoastaan hyppytaajuuden $1 \cdot 10^{15}$ 1/s data.

Alkuvarauksen +4 simulaatiot				
Fragmentti	0 varausta	1 varaus	2 varausta	3 varausta
H	6363	7019	0	0
C	2046	5132	0	0
CH	563	368	0	0
C ₂	353	735	0	0
C ₂ H	51	94	0	0
S	228	596	0	0
C ₃	9	21	0	0
CS	201	335	3	0
C ₂ S	76	611	130	0
C ₂ HS	3	13	0	0
C ₃ S	14	110	0	0
C ₄ S	6	91	0	0
I	790	3543	0	0
CI	500	1361	82	0
C ₂ I	239	1803	690	0
C ₂ HI	8	149	0	0
C ₃ I	7	26	0	0
CIS	121	1919	497	0
C ₂ IS	37	1260	31	3
C ₂ HIS	0	52	0	0
C ₃ IS	10	350	12	1
C ₃ HIS	0	13	0	0
C ₄ IS	0	53	0	0
C ₂ I ₂ S	2	251	212	31
C ₃ I ₂ S	0	69	9	0

Alkuvarauksen +2 simulaatiot			
Fragmentti	0 varausta	1 varaus	2 varausta
H	12015	1026	0
C	2724	2008	0
CH	5848	1749	0
C ₂	330	417	0
C ₂ H	581	638	0
C ₂ H ₂	360	247	0
S	540	481	0
C ₃	12	18	0
C ₃ H	31	50	0
C ₃ H ₂	22	9	0
CS	304	398	0
C ₂ S	129	433	10
C ₂ HS	114	213	0
C ₃ S	17	84	0
C ₃ HS	20	54	0
C ₃ H ₂ S	5	10	0
C ₄ S	14	65	0
C ₄ HS	27	103	0
C ₄ H ₂ S	19	71	0
I	2781	4406	0
CI	1140	2089	29
C ₂ I	381	2099	77
C ₂ HI	533	1857	0
C ₃ I	8	58	0
C ₃ HI	32	261	0
C ₃ H ₂ I	21	97	0
CIS	664	3228	244
C ₂ IS	189	1842	15
C ₂ HIS	152	1312	0
C ₃ IS	20	553	9
C ₃ HIS	51	413	1
C ₃ H ₂ IS	0	38	0
C ₄ IS	7	143	0
C ₄ HIS	9	436	0
C ₄ H ₂ IS	0	387	0
C ₄ HI ₂	0	15	0
C ₂ I ₂ S	54	1213	223
C ₃ I ₂ S	3	338	8
C ₃ HI ₂ S	14	625	0
C ₄ I ₂ S	0	42	0
C ₄ HI ₂ S	0	94	0

2.4 Molekyylifragmenttien tuoton riippuvuus sisäenergiasta

Sisäenergia on suure, jota voidaan käyttää hyväksi mallin toimivuuden arvioinnissa. Tätä varten on olennaista tarkastella eri fragmenttien tuottojen riippuvuutta sisäenergiasta, jota kappaleessa 3.2 verrataan tutkimusdataan.

Alkuvarauksen +4 simulaatioissa saatujen fragmenttien tuotot sisäisen energian funktiona on esitetty kuvasarjassa 10 ja alkuvarauksen +2 (hyppytaajuus $1 \cdot 10^{15}$ 1/s) vastaavat on esitetty kuvasarjoissa 11 ja 12. Näiden lisäksi fragmentteja molemmista simulaatiosarjoista, joiden tuotoissa ei havaita mitään poikkeuksellista, on esitetty liitteessä 1 kuvasarjoissa 16 - 18.

Kuvissa tuotto on normitettu kaikkien fragmenttien lukumäärän avulla samoilta energiaväleiltä. Virherajoina on käytetty neliöjuurta lukumäärästä. Fragmentit, joiden absoluuttinen lukumäärä on hyvin pieni (alle 50-100, riippuen arvioitujen epävarmuuksien suuruudesta), on jätetty huomiotta. Lisäksi tuottoja tarkastellaan ainoastaan energiaan 35 eV saakka, koska sitä suuremman energian trajektoreja ei ole tarpeeksi.

2.4.1 Alkuvarauksen +4 systeemin fragmenttien tuotot

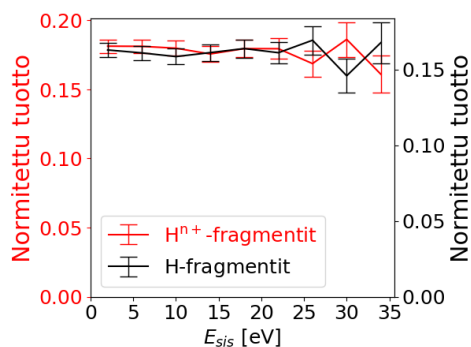
Yleisenä huomiona voidaan todeta alkuvarauksen +4 systeemin fragmenttien tuottojen riippuvan melko vähän sisäenergiasta. Useimmilla fragmenteilla ei ole havaittavissa selvää efektiä sisäenergian kasvaessa.

Vetyatomit (kuva 10a) irtoavat molekyylistä käytännössä aina, eikä sisäenergialla näytä olevan vaikutusta asiaan. Sama on havaittavissa sekä varatuilla vedyillä että varauksettomilla. Varattujen hiiliatomien (kuva 10b) tuotto pysyy lähes vakiona, mutta varauksettomien kasvaa noin 20 eV sisäenergian kohdalla. Varauksettomien hiili-atomien lukumäärä on kuitenkin paljon pienempi kuin varattujen, mikä saattaa selittää suurempaa vaihtelua tuotossa.

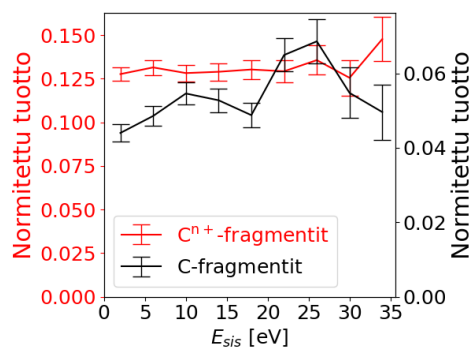
Varattujen C_4S - ja C_2I -fragmenttien tuotto (kuvat 10d ja 10f) näyttäisi laskevan

sisäenergian noustessa. Varauksettoman C₂I-fragmentin tuotto puolestaan nousee hieman sisäenergian funktiona.

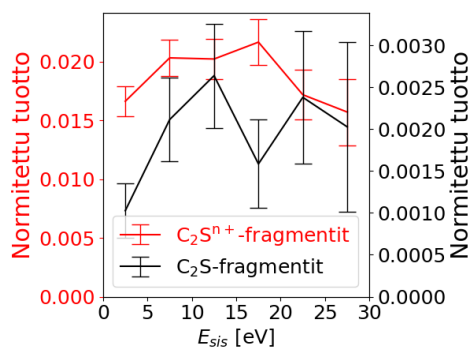
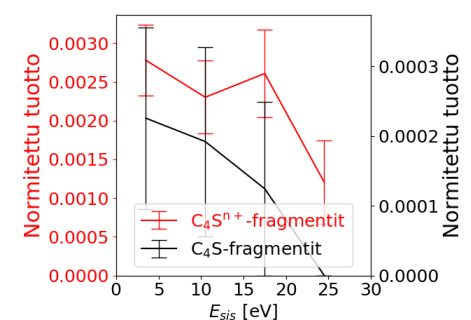
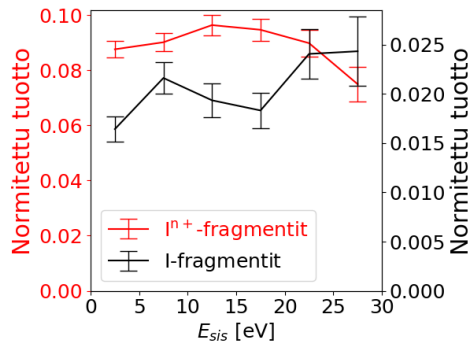
Varattujen jodien tuotto (kuva 10e) nousee noin 10-15 eV sisäenergiaan asti, mutta laskee sen jälkeen energian kasvaessa. Varauksettomien jodien tuotto puolestaan kasvaa hieman sisäenergian funktiona.



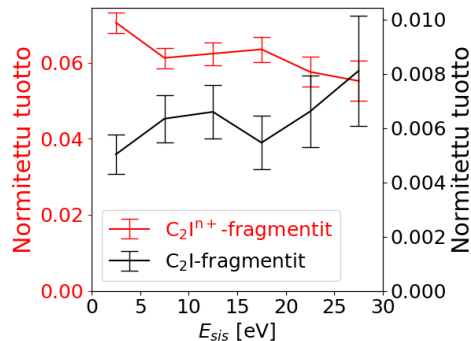
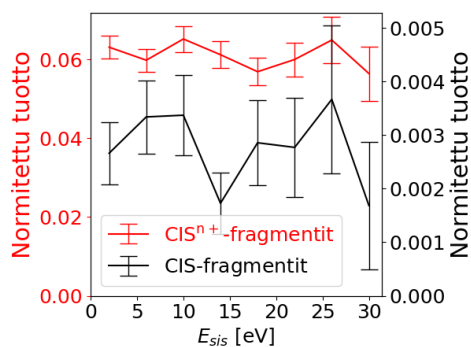
(a) Fragmentti H



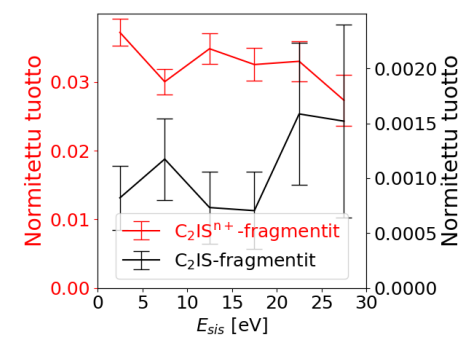
(b) Fragmentti C

(c) Fragmentti C₂S(d) Fragmentti C₄S

(e) Fragmentti I

(f) Fragmentti C₂I

(g) Fragmentti CIS

(h) Fragmentti C₂IS

Kuva 10: Alkuvarauksen +4, hyppytaajuuden $1 \cdot 10^{15}$ 1/s ja lisätyn sisäenergian 0 eV simulaatioissa syntyneiden fragmenttien tuottoja sisäenergian funktiona.

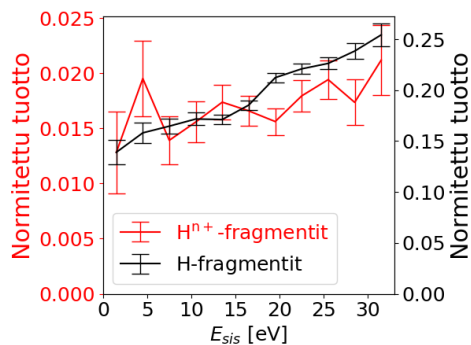
2.4.2 Alkuvarauksen +2 systeemin fragmenttien tuotot

Alkuvarauksen +2 simulaatioissa fragmenttien tuottojen ja sisäenergian välillä on nähtävissä huomattavasti suurempi riippuvuus. Yksittäisten kevyiden atomien tuotto näyttää riippuvan selvästi sisäisestä energiasta. Vedyllä (kuva 11a) on havaittavissa lähes lineaarinen kasvu tuotossa sisäenergian kasvaessa, kun taas hiilellä (kuva 11b) tuoton kasvu on silmämääräisesti eksponentiaalinen. Myös C_2 -fragmentilla (kuva 11d) on havaittavissa lähes eksponentiaalinen kasvu. Sekä varauksellisten että varauksettomien fragmenttien tuotot näyttävät riippuvan energiasta samalla tavalla näissä.

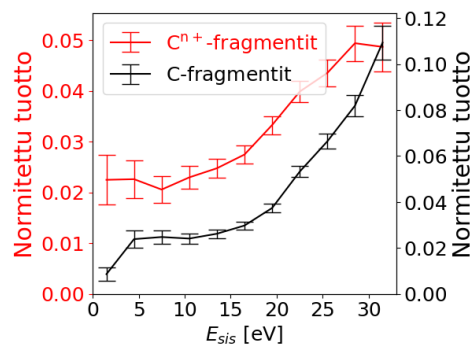
Tuoton kasvua sisäenergian funktiona on havaittavissa samaan tapaan myös S-, CS- ja C_3IS -fragmenteilla (kuvat 11e,11f ja 12c), sekä varatuilla että varauksettomilla fragmenteilla. Jodin tapauksessa (kuva 12a) mielenkiintoisesti varatun tuotto laskee sisäenergian kasvaessa, mutta varauksettomalla on havaittavissa tuoton lineaarinen kasvu. Varauksettomien vetyjen irtoamisen energiariippuvuutta onkin havaittu esimerkiksi oikeilla tiofeeni-molekyyleillä (C_4H_4S) [18].

Fragmenttien, joiden osana on vetyä, tuotto (kuvat 11c,12b,12d ja 12e) laskee energian funktiona, mikä onkin odotettua vedyn näyttäessä vastakkaista efektiä.

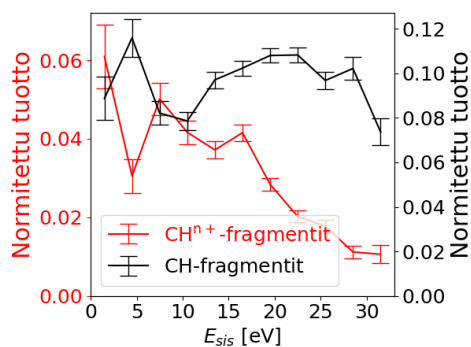
Yleisesti siis pienempien fragmenttien tuotot kasvavat pääosin sisäenergian funktiona, mikä on odotettuakin, koska luonnollisesti suuremman määrän sisäenergiaa voidaan olettaa myös hajottavan molekyylin todennäköisemmin. Tämä on myös havaittavissa suurten fragmenttien tuoton laskusta energian funktiona (esimerkiksi C_2I_2S , kuva 12f), vaikka monen suuren fragmentin tapauksessa vetyjen irtoaminenkin laskee tuottoa. Taulukosta III voidaan kuitenkin nähdä, että vedyttömiä suurien fragmenttien lukumäärä on ylipäänsä pieni, joten voidaan olettaa molekyylin hajoavan useampiin osiin vetyjen irrotessa.



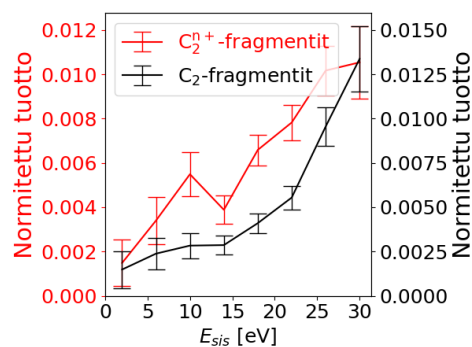
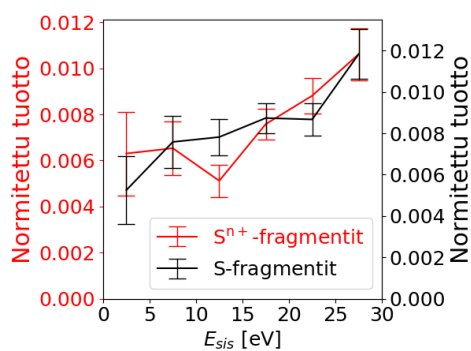
(a) Fragmentti H



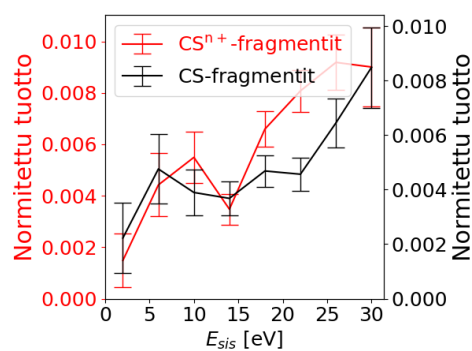
(b) Fragmentti C



(c) Fragmentti CH

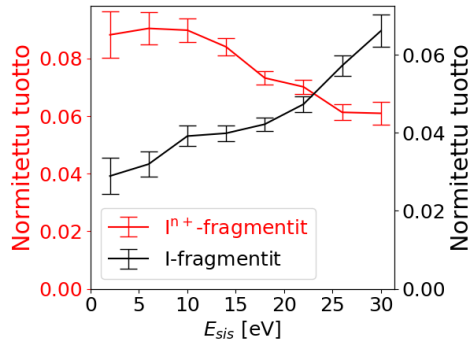
(d) Fragmentti C₂

(e) Fragmentti S

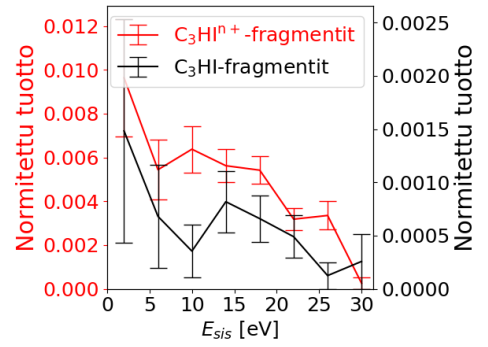
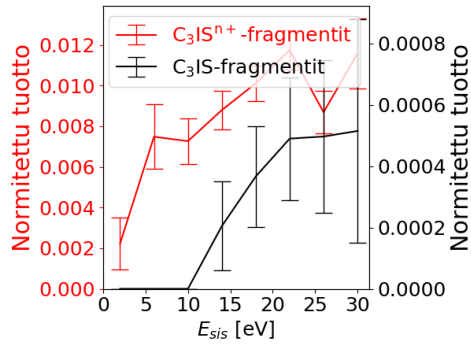
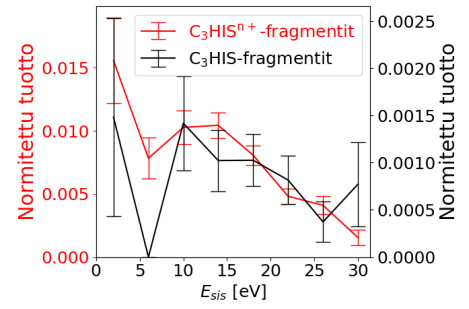
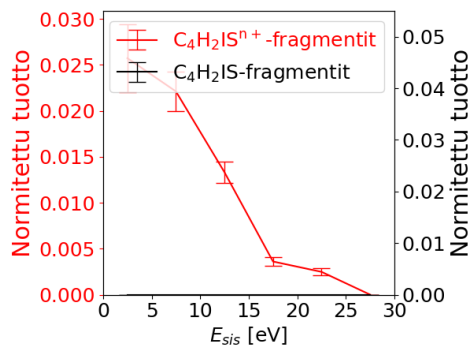
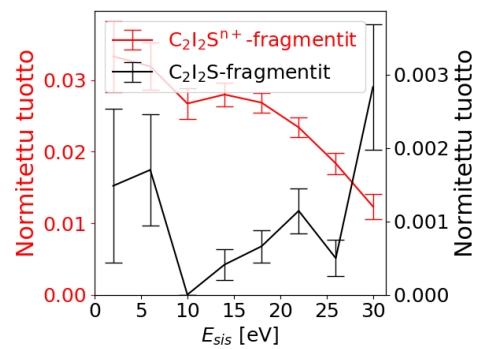


(f) Fragmentti CS

Kuva 11: Alkuvarauksen $+2$, hyppytaajuuden $1 \cdot 10^{15}$ 1/s ja lisätyn sisäenergian 10 eV simulaatioissa syntyneiden fragmenttien tuottoja sisäenergian funktiona.



(a) Fragmentti I

(b) Fragmentti C₃HI(c) Fragmentti C₃IS(d) Fragmentti C₃HIS(e) Fragmentti C₄H₂IS(f) Fragmentti C₂I₂S

Kuva 12: Alkuvarauksen +2, hyppytaajuuden $1 \cdot 10^{15}$ 1/s ja lisätyn sisäenergian 10 eV simulaatioissa syntyneiden fragmenttien tuottoja sisäenergian funktiona.

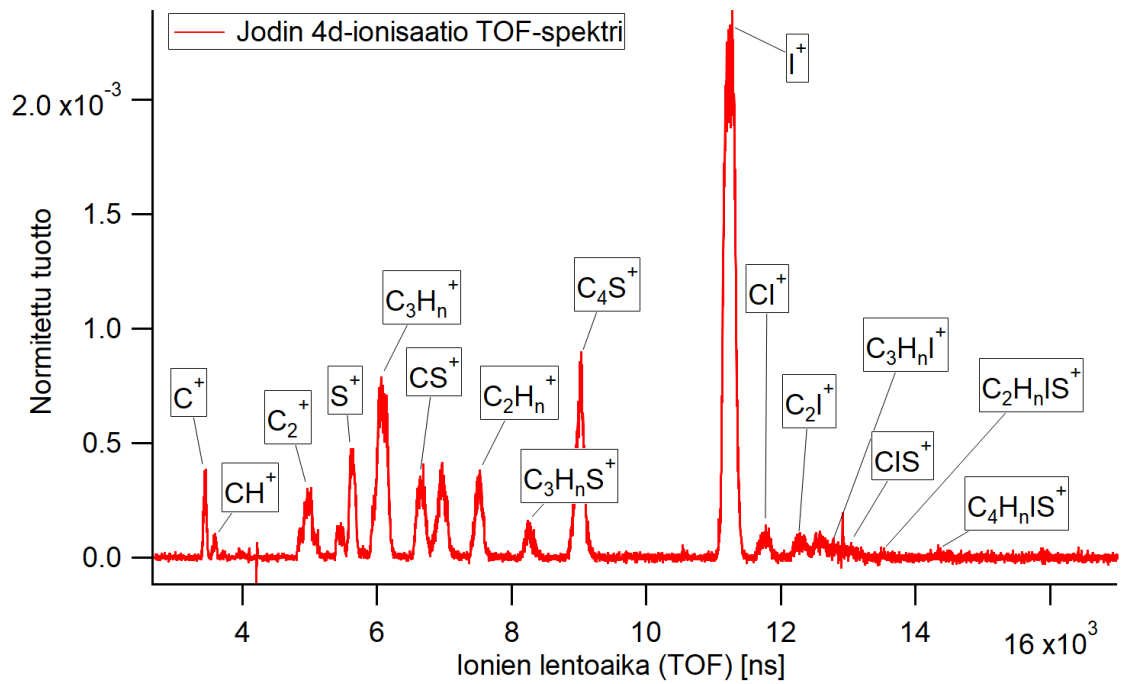
3 Tulosten vertailu

Tutkielmassa rakennetun molekyylimallin antamien tulosten arvioimiseksi on tehtävä vertailua niiden ja oikeiden molekyylien hajoamisen tuottaman datan välillä. Tätä varten tarkastellaan ionien lentoaikadataa (TOF) diiodiofeenin hajomisesta sekä jodin 4d-ionisaation (+2 varaus) että 3d-ionisaation (+3,+4 varaus) seurauksena.

Data molekyylien hajoamisesta on saatu käyttäen FinEstBeAMS:ia Lundissa, Ruotsissa sijaitsevassa MAX IV -laboratoriossa. Molekyylien ionisaatio tapahtui käyttäen säädettävää intensiteettistä röntgensäteilyä synkrotronista. Ionisaation seurauksena hajonneiden molekyylien fragmentit havaittiin käyttäen lentoaikaspektrometriä (TOF-spektrometri). Tarkempia tietoja tutkimuslaitteistosta on saatavilla Kooser *et al.* artikkelista [19].

3.1 Fragmenttien jakauma

Ionien lentoaikadatasta etsittiin tämän tutkielman simulaation tuottamat fragmentit, joiden tuoton suhde kaikkiin ioneihin laskettiin. Jodin 4d-ionisaation TOF-spektri on esitetty kuvassa 13, johon on myös merkitty kaikki vertailussa käytettyjä ioneja vastaavat piikit.



Kuva 13: Jodin 4d-ionisaation TOF-spektri.

Fragmenttien suhteelliset tuotot on esitetty taulukossa IV sekä tutkimusdatalle että simulaatioiden tuottamalle datalle.

Simulaatioiden datasta on poistettu yksittäisten vetyjen (ionit ja atomit) tuotot, sillä niitä vastaava lentoaika oli vertailudatan mitta-alueen ulkopuolella. Tämän lisäksi simulaatiodatasta vertailun kohteena on ainoastaan $+1$ varauksen fragmentit, koska vertailudata koostuu luonnollisesti ainoastaan ioneista. Ainoastaan suurempi alkuvarauksen $+2$ simulaatioasetti (hyppytaajuus $1 \cdot 10^{15}$) on osana vertailua.

Taulukko IV: Alkuvarausten +4 ja +2 (hyppytaajuus $1 \cdot 10^{15}$) simulaatioissa syntyneiden +1 varauksellisten ionien suhteelliset osuudet verrattuna tutkimusdataan.

Alkuvarauksen +4 simulaatiot			Alkuvarauksen +2 simulaatiot		
Fragmentti	Simulaatiodata	Tutkimusdata	Fragmentti	Simulaatiodata	Tutkimusdata
C	0.2704	0.3396	C	0.0703	0.0168
CH	0.0194	0.0119	CH	0.0612	0.0047
C ₂	0.0387	0.0610	C ₂ H _{0,1,2}	0.0456	0.0438
C ₂ H	0.0050	0.0070	S	0.0168	0.0448
S	0.0314	0.1071	C ₃ H _{0,1}	0.0024	0.1312
C ₃	0.0011	0.0229	CS	0.0139	0.0467
C ₃ H	0.0003	0.0034	C ₂ H _{0,1} S	0.0226	0.0427
CS	0.0177	0.0061	C ₃ H _{0,1} S	0.0048	0.0168
C ₂ H _{0,1} S	0.0329	0.0014	C ₄ H _{0,1,2} S	0.0084	0.1084
C ₃ H _{0,1} S	0.0060	0.0014	I	0.1542	0.4254
C ₄ H _{0,1} S	0.0050	0.0026	CI	0.0731	0.0152
I	0.1867	0.3298	C ₂ H _{0,1} I	0.1384	0.0135
CI	0.0717	0.0035	C ₃ H _{0,1,2} I	0.0146	0.0053
C ₂ I	0.0950	-	CIS	0.1130	0.0051
C ₂ HI	0.0079	-	C ₂ H _{0,1} IS	0.1104	0.0008
C ₃ I	0.0014	-	C ₃ IS	0.0194	-
C ₃ HI	0.0003	-	C ₃ HIS	0.0145	-
CIS	0.1011	-	C ₃ H ₂ IS	0.0013	-
C ₄ I	0.0001	-	C ₄ H _{0,1,2} IS	0.0338	0.0009
C ₄ HI	0.0001	-	C ₄ HI ₂	0.0005	-
C ₂ IS	0.0664	-	C ₂ I ₂ S	0.0424	-
C ₂ HIS	0.0027	-	C ₃ I ₂ S	0.0118	-
C ₃ IS	0.0184	-	C ₃ HI ₂ S	0.0219	-
C ₃ HIS	0.0007	-	C ₄ I ₂ S	0.0015	-
C ₄ IS	0.0028	-	C ₄ HI ₂ S	0.0033	-
C ₄ I ₂	0.0001	-			
C ₂ I ₂ S	0.0132	-			
C ₃ I ₂ S	0.0036	-			

Alkuvarauksen +4 simulaatiot tuottivat vertailudataa huomattavasti vähemmän pienimassaisia ioneja kuten C^+ , C_2^+ ja S^+ . Myös I^+ -ionin tuotto on vertailudatassa lähes kaksi kertaa suurempi kuin simulaatioissa. Suurimassaisia ioneja ($m > 150$ u) puolestaan syntyy simulaatioissa, mutta niitä ei ole havaittavissa vertailudatassa. Oikeissa hajoamistapahtumissa selvästi alkuperäiset molekyylit hajoavat siis pienemmiksi palasiksi.

Alkuvarauksen +2 simulaatioissa Kokonaisia +2 varattuja diioditiofeeni-ioneja löytyi 4d-ionisaation vertailudatasta, mutta simulaatiot eivät tuottaneet niitä ollenkaan. Simulaatiot näyttävät tuottavat todellista huomattavasti laajemman jakouman ioneja, pienimassaisia ioneja ($m < 35$ u) syntyy enemmän, samoin kuin hyvin suurimassaisia ($m > 200$ u). Vertailudatasta yli 40% on jodi-ioneja, kun taas simulaatioiden tuotto on sen osalta vain noin 15%.

3.2 Fragmenttien tuotot energian funktiona

Edellisessä kappaleessa esitetystä ionien lentoaikadatasta (TOF-spektri kuvassa 13) on valittu tarkempaa vertailua varten myös simulaatiodatan tapaan yksittäisten ionien tuottoja energian funktiona. Olennainen ero simulaatioiden tuottamaan dataan on kuitenkin se, että ionien lentoaikaa mitattaessa ei voida havaita suoraan mitään sisäenergiaan verrattavissa olevaa suuretta.

Sisäenergiaa arvioidaan yksinkertaistetusti auger-ilmiössä vapautuneen energian avulla, joka puolestaan saadaan määritettyä samassa hajoamistapahtumassa irronneen elektronin kineettisestä energiasta. Tätä varten oletetaan auger-ilmiössä vapautuneen energian maksimin vastaavan suurimpia havaittuja kineettisiä energioita (40 eV) ja, että tätä pienempi kineettinen energia tarkoittaa suureiden erotuksena olevan määrän olevan sisäenergiaa. Tällöin siis sisäenergia saadaan alla esitetyllä tavalla.

$$E_{\text{sis,vertailu}} = 40 \text{ eV} - E_{\text{kin,e}} \quad (11)$$

Tutkimusdatasta vertailuun käytetään kuvassa 13 esitettyä 4d-ionisaation tuottamaa dataa, josta on valittu sellaiset ionit, joita on syntynyt myös simulaatioissa ja joiden tuotossa sisäenergian funktiona on havaittavissa muutosta energian kasvaessa. Tuotot on normitettu käyttäen kaikkien ionien lukumäärää samoilla energiaväleillä ja virherajoina on esitetty neliöjuuri lukumäärästä.

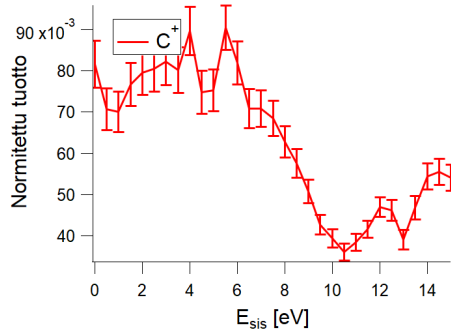
Simulaatiodatasta 4d-ionisaatiota vastaavat +2 alkuvarauksen setit, joista vertailuun käytetään jo aiemmin kappaleessa 2.4.1 (kuvat 11 ja 12) esitetyt ionien tuotot sisäenergian funktiona hyppytaajuuden $1 \cdot 10^{15}$ 1/s simulaatioissa. Ainoastaan varattuja fragmentteja voidaan verrata, koska lentoaikadata luonnollisesti sisältää ainoastaan varattuja hiukkasia. Vertailudatan ja simulaatioiden versiot ionien tuotoista sisäenergian funktiona on esitetty rinnakkain kuvissa 14 ja 15.

Vertailudatassa energia-asteikko on huomattavasti pienempi (0-15 eV) kuin simulaatiodatan (0-40 eV, 0-35 eV), mitä havainnollistamaan kuviin on merkitty katkoiviivalla 15 eV sijainti. Simulaatiot näyttävät siis tuottavan todellista huomattavasti suurempienergisiiä fragmentteja, vaikka kineettisen energian muutos sisäenergiaksi vertailudatan tapauksessa voikin vaikuttaa asiaan.

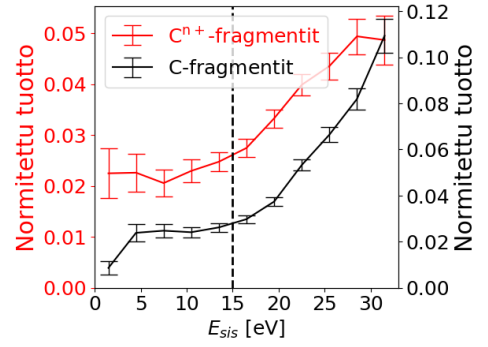
Kuvasetissä 14 on havaittavissa käytännössä päinvastainen efekti tuotoissa simulaation ja vertailudatan välillä fragmenttien C^+ (kuvat 14a,14b), C_2^+ (kuvat 14c,14d) ja S^+ (kuvat 14d,14e) tapauksessa. Simulaatioiden tuotto nousee energian funktiona, mutta vertailudatan laskee. $\text{C}_2\text{H}_n\text{S}^+$ -fragmentin tuotto (kuvat 14f,14g) puolestaan vastaa suurinpiirtein toisiaan vertailtavissa olevalla energiavälillä.

Kuvasetissä 15 $\text{C}_4\text{H}_n\text{S}^+$ -fragmentin tuotossa (kuvat 15a,15b) on havaittavissa huomattava kasvu molemmissa dataseteissä yli 5 eV energioilla, mutta simulaatiot tuottivat erittäin vähän kyseisiä fragmentteja tehden vertailun hankalaksi. Yksittäisen jodin tuotto (kuvat 15c,15d) näyttää laskevan energian funktiona sekä simu-

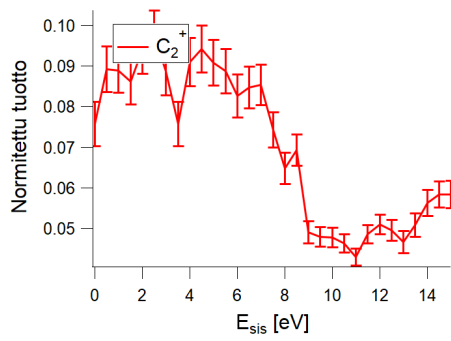
laatiotatassa että vertailutatassa. C_2I^+ -fragmentin tuotossa on havaittavissa pieni kasvu energian funktiona molemmissa dataseteissä. CIS^+ -fragmentin tuotto simulaatioissa pysyy melko tasaisena, eikä vertailutatassa näkyvää kasvua välillä 2-8 eV ole havaittavissa.



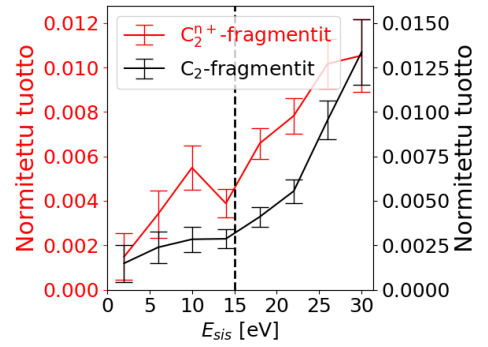
(a) Tutkimusdata



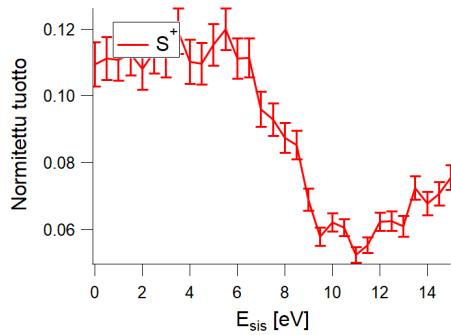
(b) Simulaatiodata



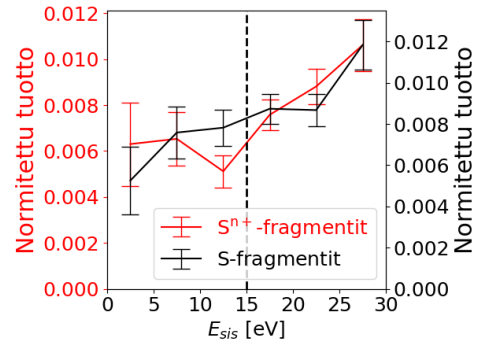
(c) Tutkimusdata



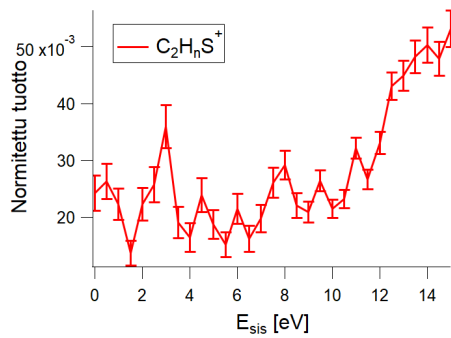
(d) Simulaatiodata



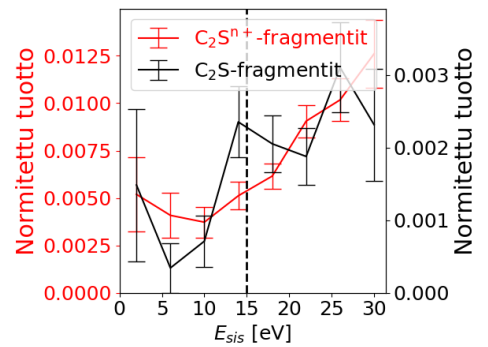
(e) Tutkimusdata



(f) Simulaatiodata

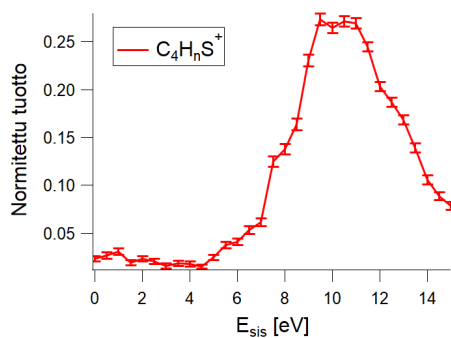


(g) Tutkimusdata

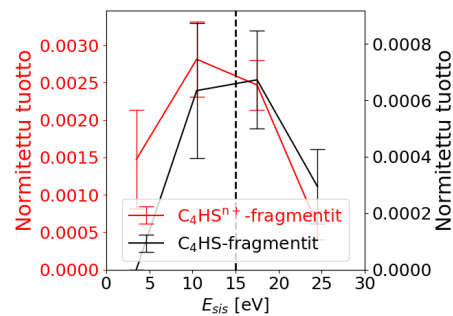


(h) Simulaatiodata

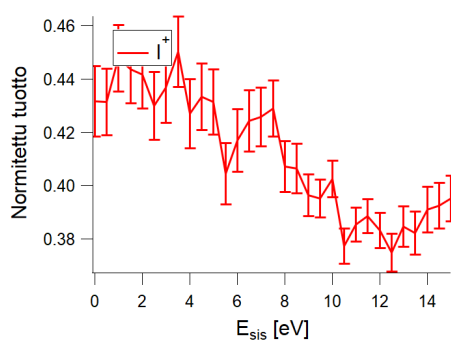
Kuva 14: Fragmenttien tuottoja sisäenergian funktiona vertailudatassa ja simulaatioissa (varaus +2, hyppytaajuus $1 \cdot 10^{15}$).



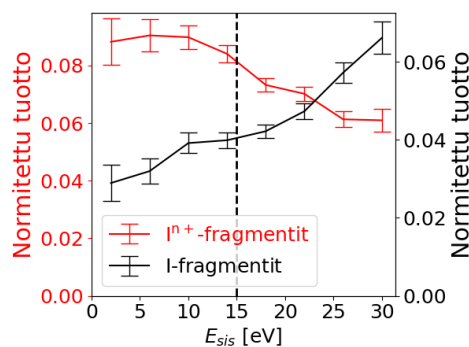
(a) Tutkimusdata



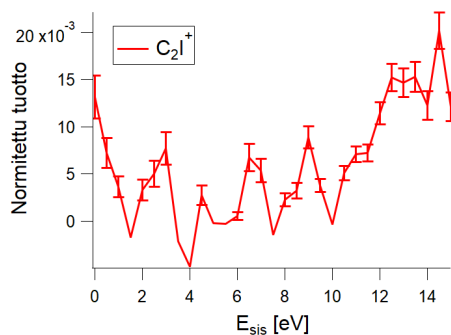
(b) Simulaatiodata



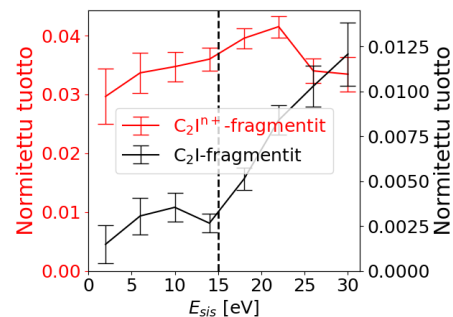
(c) Tutkimusdata



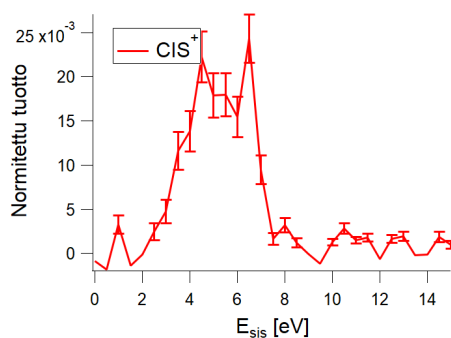
(d) Simulaatiodata



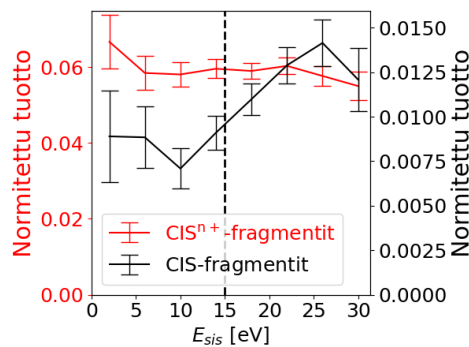
(e) Tutkimusdata



(f) Simulaatiodata



(g) Tutkimusdata



(h) Simulaatiodata

Kuva 15: Fragmenttien tuottoja sisäenergian funktiona vertailudatassa ja simulaatioissa (varaus +2, hyppytaajuus $1 \cdot 10^{15}$).

4 Yhteenveto

Tutkielman tavoitteena oli luoda yksinkertainen malli, jolla molekyylin hajoamista olisi mahdollista tutkia statistisesti. Mallin rakentaminen onnistui suunnitelmien mukaan ilman suuria teknisiä ongelmia. Alkuperäinen idea myös kehittyi prosessin aikana ja malliin lisättiin ominaisuuksia kuten atomit toisistaan erossa pitävä jousipotentiaali, kun välituloksia tarkasteltiin.

Rakennettu malli toimii fysikaalisesti järkevästi. Energian lisääminen systeemiin aiheuttaa molekyylin hajoamista useammaksi palaseksi ja kappaleet käyttäytyvät muutenkin odotusten mukaisesti. Mallin rakentamisen voidaan siis todeta onnistuneen.

Miten mallin tulokset sitten vertautuivat todellisten molekyylien hajomisten tuotamiin, huomioiden rajut yksinkertaistukset mallin alkuoletuksissa todelliseen hajoamistapahtumaan nähden? Simulaatioiden tuottamat fragmenttijakaumat olivat pääosin paljon laajempia (enemmän uniikkeja fragmentteja) kuin oikeiden molekyylien hajoamisessa syntyneet ja esimerkiksi jodien-ionien ja pienimassaisten ionien tuotot olivat todellista pienempiä. Fragmenttien tuotossa sisäenergian funktiona oli havaittavissa huomattavia eroja monen verratun fragmentin tapauksessa (C^+ , C_2^+ , CIS^+ ja S^+), mutta toisissa muutokset tuotossa olivat samansuuntaisia sekä simulaatiodatassa että vertailudatassa.

Voidaan siis todeta mallin tuottavan nykymuodossaan tuloksia, jotka poikkeavat huomattavasti kokeellisista, mikä toisaalta olikin odotettua. Pro Gradu -tutkielman laajuuteen ei mahtunut syvempää tarkastelua mallin toiminnasta tai sen jatkokehitystä, joten on mahdotonta tehdä lopullisia johtopäätöksiä. On kuitenkin nähtävissä, että mallissa on potentiaalia, joten sen kehityksen jatkaminen jossain muodossa vaikuttaa järkevältä idealta.

Mallissa on useita parametrejä, joita ei tutkielmaa varten ollut mahdollista tutkia tarkasti. On luonnollista, että ensimmäinen parannuskohde olisi pyrkiä säätämään

parametreja ja tarkastelemaan sen vaikutusta tuloksiin. Esimerkiksi Alkuvarauksen +2 systeemin fragmenttituotto olisi saattanut vastata paremmin vertailudataa pienemmällä määrällä lisättyä lisäenergiaa, koska tällä kertaa jakauma oli liian laaja.

Parametrien arvojen testauksessa huomioon tulee kuitenkin ottaa muunmuassa se, ettei parametrejä suoraan soviteta vertailudataan, vaan tulosten vertailu tehdään esimerkiksi kokonaan riippumattomalla datalla. On mahdollista, että paremilla valinnoilla tulokset ovat lähempänä oikeiden molekyylien hajoamisen tuottamia vastaavia. Parametrien testaaminen vaatii myös itse koodin optimointia, koska nykyisellään laskentatehoa vaaditaan liikaa suuren parametriyhdistelmien kokeilemiseen.

Varausten hyppiminen on toteutettu hyvin yksinkertaisesti (pitkästi tutkielman tavoitteen mukaisesti), mikä voi tarkoittaa sitä, ettei malli kykene vastaamaan monimutkaista todellisuutta. Hyppyjä olisi mahdollista rajoittaa esimerkiksi yksinkertaisen Coulombin potentiaalin avulla asettamalla raja sille, paljonko varausten liikuminen voi "luoda" energiaa.

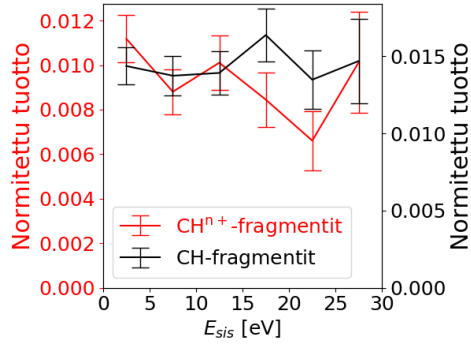
Atomien välisten sidosten rikkoutuminen ja muodostuminen simulaation aikana voisi myös olla yksi tapa saada malli vastaamaan paremmin todellisuutta. On hyvin epäfysikaalista, että atomit ovat kytkettyjä toisiinsa alkugeometrian mukaisesti eikä systeemi kehity ajan kuluessa. Ongelman ratkaiseminen voi kuitenkin olla haastavaa toteuttaa niin, että malli pysyy siitä huolimatta yksinkertaisena.

Viitteet

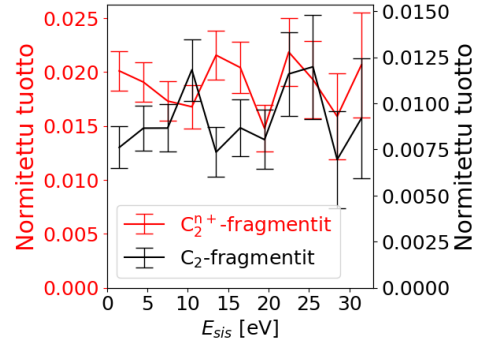
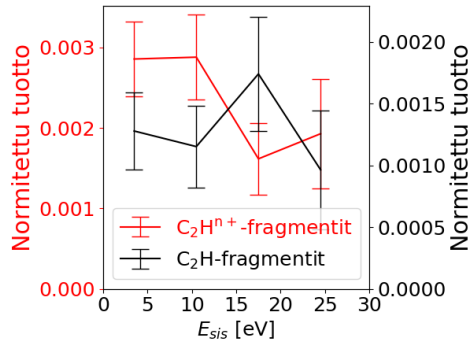
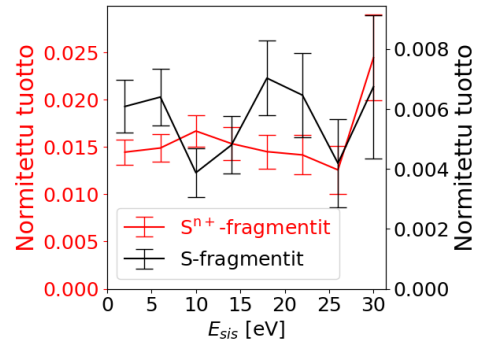
- [1] A. Laaksonen ja Y. Tu, kirjassa *Molecular Dynamics*, Vol. 7 of *Theoretical and Computational Chemistry*, toimittanut P. B. Balbuena ja J. M. Seminario (Elsevier, 1999), pp. 1–29.
- [2] D. O’Carroll ja N. J. English, *Journal of Chemical Physics* **153**, (2020).
- [3] L. H. Krarup, I. T. Christensen, L. Hovgaard ja S. Frokjaer, *Pharmaceutical Research* **15**, (1998).
- [4] S. J. Plimpton, D. Perez ja A. F. Voter, *Journal of Chemical Physics* **153**, (2020).
- [5] T. Onishi, *Quantum computational chemistry : modelling and calculation for functional materials* (Springer, 2017).
- [6] T. Tsuneda, *Density functional theory in quantum chemistry* (Springer, 2014).
- [7] A. S. Folorunso, A. Bruner, F. m. c. Mauger, K. A. Hamer, S. Hernandez, R. R. Jones, L. F. DiMauro, M. B. Gaarde, K. J. Schafer ja K. Lopata, *Phys. Rev. Lett.* **126**, 133002 (2021).
- [8] R. Iftimie, P. Minary ja M. E. Tuckerman, *Proceedings of the National Academy of Sciences of the United States of America* **102**, 6654 (2005).
- [9] P. M. Morse, *Phys. Rev.* **34**, 57 (1929).
- [10] National Center for Biotechnology Information, PubChem Compound Summary for CID 69368, 2,5-Diiodothiophene, 2021, https://pubchem.ncbi.nlm.nih.gov/compound/2_5-Diiodothiophene, Luettu viimeksi 2021-08-04.
- [11] CIAAW, Atomic weights of the elements, 2021, <https://www.ciaaw.org>, Luettu viimeksi 2021-08-04.
- [12] NIST, Atomic Reference Data for Electronic Structure Calculations, Electronic Configurations of the Elements, 2021, <https://www.nist.gov/pml/atomic-reference-data-electronic-structure-calculations/atomic-reference-data-electronic-8>, Luettu viimeksi 2021-08-04.
- [13] T. L. Cottrell, *Properties of atoms, radicals, and bond* **372**, (1966).
- [14] B. D. Darwent, *Bond Dissociation Energies in Simple Molecules*, 1970.
- [15] M. Diem, *Modern vibrational spectroscopy and micro-spectroscopy : theory, instrumentation, and biomedical applications*, 2015.
- [16] M. A. Cuendet ja W. F. Van Gunsteren, *Journal of Chemical Physics* **127**, (2007).
- [17] M. Fujimoto, *Physics of classical electromagnetism* (Springer, 2007), p. 58.

- [18] E. Kukk, D. T. Ha, Y. Wang, D. G. Piekarski, S. Diaz-Tendero, K. Kooser, E. Itälä, H. Levola, M. Alcamí, E. Rachlew ja F. Martín, *Physical review. A, Atomic, molecular, and optical physics* **91**, (2015).
- [19] K. Kooser, A. Kivimäki, P. Turunen, R. Pärna, L. Reisberg, M. Kirm, M. Valden, M. Huttula ja E. Kukk, *Journal of synchrotron radiation* **27**, 1080 (2020).

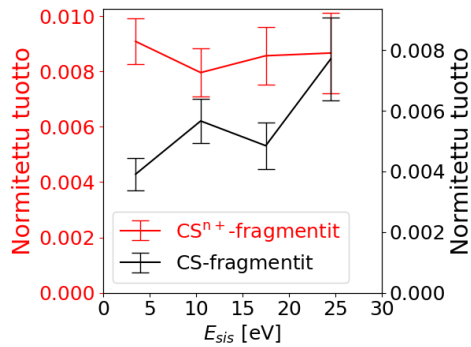
5 Liite 1: Lisää ionituottokuvaajia



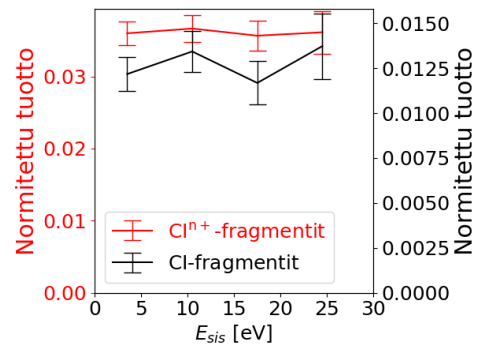
(a) Fragmentti CH

(b) Fragmentti C₂(c) Fragmentti C₂H

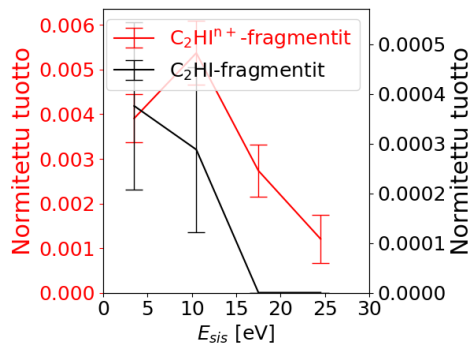
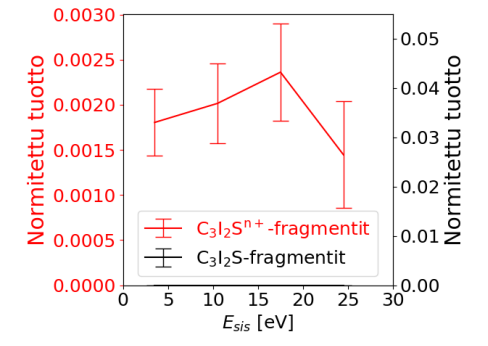
(d) Fragmentti S



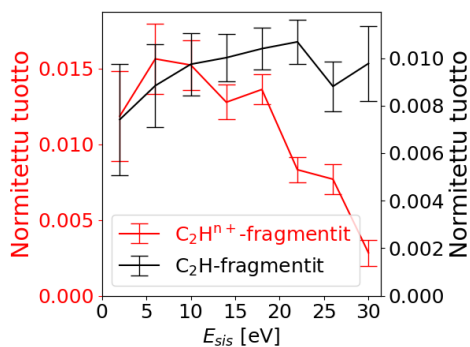
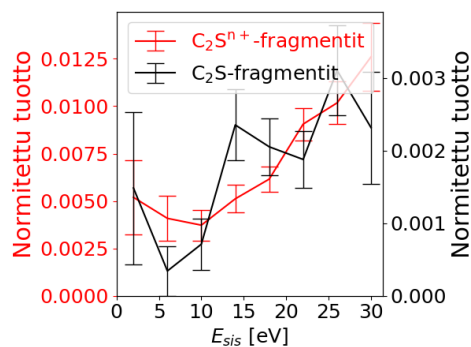
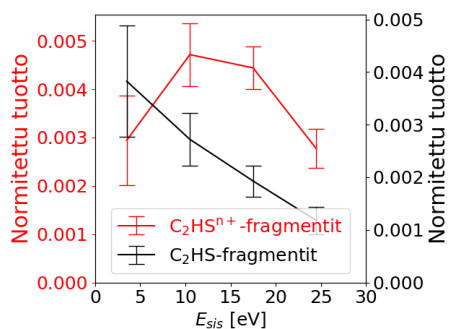
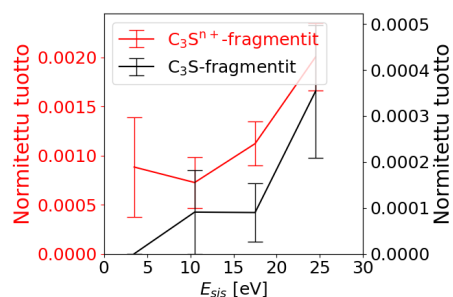
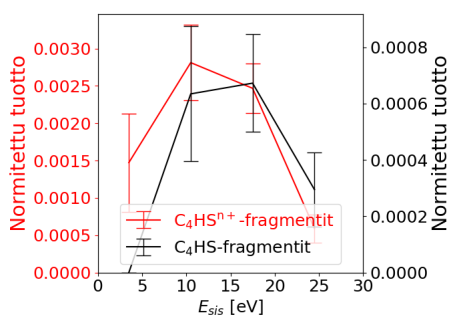
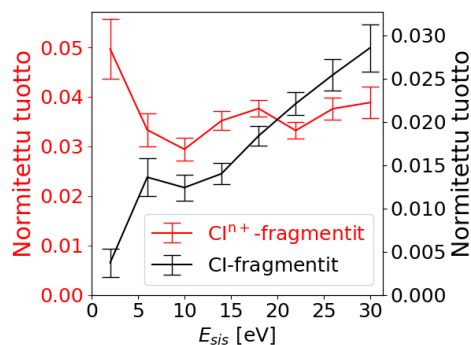
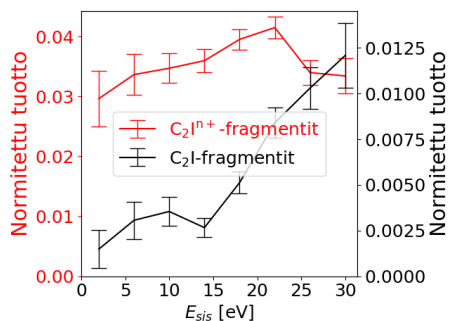
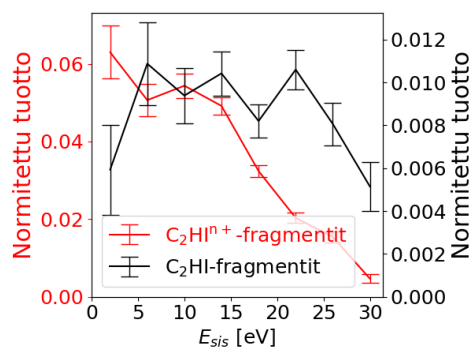
(e) Fragmentti CS



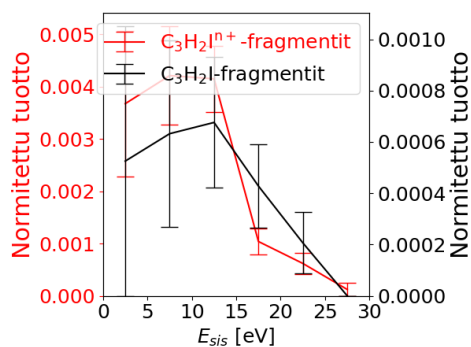
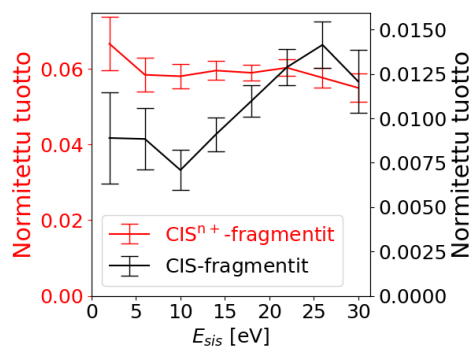
(f) Fragmentti CI

(g) Fragmentti C₂HI(h) Fragmentti C₃I₂S

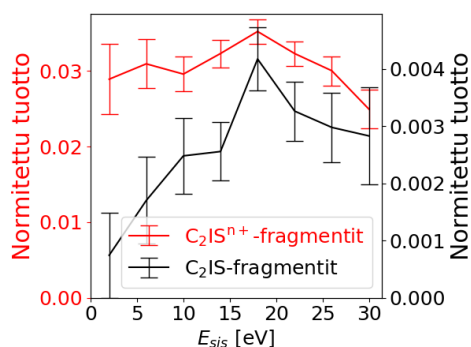
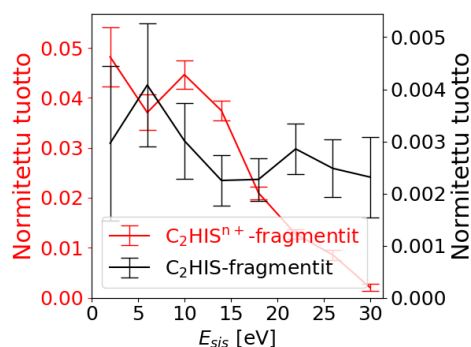
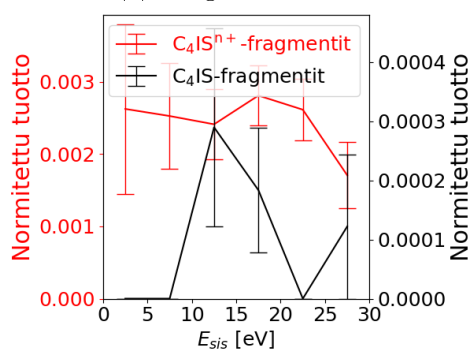
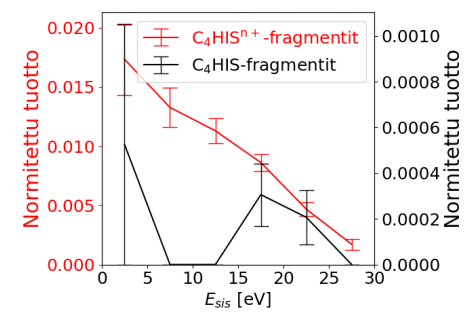
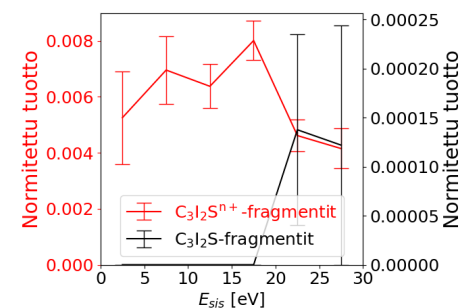
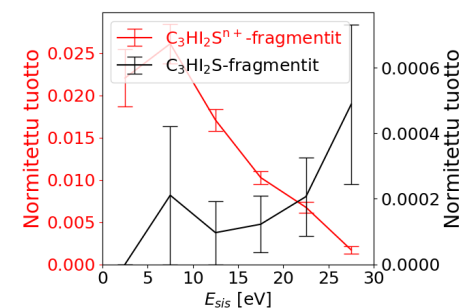
Kuva 16: Alkuvarauksen +4, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 0 eV simulaatioissa syntyneiden fragmenttien tuottoja sisäenergian funktiona.

(a) Fragmentti C_2H (b) Fragmentti C_2S (c) Fragmentti C_2HS (d) Fragmentti C_3S (e) Fragmentti C_4HS (f) Fragmentti Cl (g) Fragmentti C_2I (h) Fragmentti C_2HI

Kuva 17: Alkuvarauksen $+2$, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 10 eV simulaatioissa syntyneiden fragmenttien tuottoja sisäenergian funktiona.

(a) Fragmentti C_3H_2I 

(b) Fragmentti CIS

(c) Fragmentti C_2IS (d) Fragmentti C_2HIS (e) Fragmentti C_4IS (f) Fragmentti C_4HIS (g) Fragmentti C_3I_2S (h) Fragmentti C_3HI_2S

Kuva 18: Alkuvarauksen +2, hyppytaajuuden $1 \cdot 10^{15}$ 1/s, lisätyn sisäenergian 10 eV simulaatioissa syntyneiden fragmenttien tuottoja sisäenergian funktiona.

6 Liite 2: Simulaation koodi

```

1 #!/usr/bin/env python3
2 import sys
3 import numpy as np
4 from scipy.spatial.distance import pdist
5 from scipy.spatial.distance import euclidean
6 from scipy.spatial.distance import squareform
7 import pandas as pd
8 import matplotlib.pyplot as plt
9 import seaborn as sns
10 import datetime
11
12 # unit conversion and constants
13 eV_to_J = 1.602E-19
14 amu_to_kg = 1.660539E-27
15 A_to_m = 1E-10
16 e_to_C = 1.602E-19
17 k_e = 8.988E9 # Nm^2/C^2
18 hbar = 1.0545718E-34
19
20
21 # atom parameters and movement functions
22 class Atom:
23     def __init__(self, type, position, velocity, mass, Nel, rad,
24                 connections=0, charge=0):
25         self.type = type
26         self.position = np.array(position)
27         self.velocity = np.array(velocity)
28         self.connections = np.array(connections)
29         self.mass = mass
30         self.Nel = Nel
31         self.charge = charge
32         self.rad = rad
33
34     def move(self, dt):
35         self.position += self.velocity * dt
36
37     def accelerate(self, force, dt):
38         self.velocity += force * dt / self.mass
39
40 # system to hold general parameters
41 class System:
42     def __init__(self, hop_limit, internal_bond_energy, k,
43                 hop_frequency, frag_limiter=2.0):
44         self.hop_limit = hop_limit * A_to_m
45         self.k = k
46         self.hop_frequency = hop_frequency
47         self.internal_bond_energy = internal_bond_energy * eV_to_J
48         self.frag_limiter = frag_limiter * A_to_m
49
50 # accelerate atoms
51 def update_velocities(atoms, forces, dt):

```

```

51     for i in range(len(atoms)):
52         atom = atoms[i]
53         force = forces[i, :]
54         atom.accelerate(force, dt)
55
56 # move atoms
57 def move_all_atoms(atoms, dt):
58     for atom in atoms:
59         atom.move(dt)
60
61
62 # calculating forces for each atom for integration
63 def calc_forces(atoms, system, eq_distances, connections=0):
64     n_atoms = len(atoms)
65     forces = np.zeros((n_atoms, 3))
66
67     positions = []
68     for atom in atoms:
69         positions.append(atom.position)
70
71     distances = pdist(np.array(positions))
72     distance_matrix = squareform(distances)
73
74     for i in range(n_atoms):
75         for j in range(i + 1, n_atoms):
76
77             atom_i = atoms[i]
78             atom_j = atoms[j]
79
80             charge1 = atom_i.charge * e_to_C
81             charge2 = atom_j.charge * e_to_C
82             Dee, Eq, m, k, spring_r = get_params(system, atom_i,
atom_j)
83
84             Re = eq_distances[i][j]
85             x = distance_matrix[i][j]
86
87             coulomb_force = k_e * (charge1 * charge2) / (x * x) *
f_vector(atom_i, atom_j, x)
88
89             forces[i] -= coulomb_force
90             forces[j] += coulomb_force
91
92             if connections:
93                 if i not in atom_j.connections:
94                     if x < spring_r:
95                         spring_force = -k * (spring_r - x) *
f_vector(atom_i, atom_j, x)
96
97                         forces[i] += spring_force
98                         forces[j] -= spring_force
99                 continue
100
101     b = Eq / 2 / hbar * np.sqrt(2 * m / Dee)
102

```

```

103         morse_force = 2 * b * Dee * np.exp(b * (Re - x)) * (1 -
np.exp(b * (Re - x))) * f_vector(atom_i, atom_j, x)
104
105         forces[i] += morse_force
106         forces[j] -= morse_force
107
108     return forces
109
110
111 # leapfrog integrator, connections=1 enables potential only for
bonded atoms
112 def leapfrog_integrator(atoms, system, eq_distances,
trajectory_file, dt, time, connection=0):
113     steps = int(time / dt)
114
115     forces = calc_forces(atoms, system, eq_distances, connection)
116
117     update_velocities(atoms, forces, 0.5 * dt)
118
119     for i in range(steps):
120
121         if i % 200 == 0:
122             write_trajectory(trajectory_file, atoms)
123
124             move_all_atoms(atoms, dt)
125
126             if i % system.hop_frequency == 0:
127                 atoms = hop_charges(atoms, system)
128
129                 forces = calc_forces(atoms, system, eq_distances,
connection)
130
131                 update_velocities(atoms, forces, dt)
132
133                 update_velocities(atoms, forces, -0.5 * dt)
134
135     return atoms
136
137
138 # return parameters for specific atom pair
139 def get_params(system, atom1, atom2):
140     t1 = atom1.type
141     t2 = atom2.type
142     k = system.k # N/m
143     spring_r = atom1.rad + atom2.rad
144     m = 1 / ((1 / atom1.mass) + (1 / atom2.mass))
145
146     if t1 == 'I1' or t1 == 'I2':
147         Dee = 213 * 1.66033E-21 - system.internal_bond_energy
148         Eq = 0.074 * eV_to_J
149     elif t1 == 'S1':
150         Dee = 272 * 1.66033E-21 - system.internal_bond_energy
151         Eq = 0.086 * eV_to_J
152     elif 'H' in t2:
153         Dee = 411 * 1.66033E-21 - system.internal_bond_energy

```

```

154     Eq = 0.372 * eV_to_J
155     else:
156         Dee = 346 * 1.66033E-21 - system.internal_bond_energy
157         Eq = 0.185 * eV_to_J
158
159     Dee = get_charged_dee(atom1, atom2, Dee)
160     return Dee, Eq, m, k, spring_r
161
162
163 # hop charges
164 def hop_charges(atoms, system):
165     positions = []
166     for atom in atoms:
167         positions.append(atom.position)
168     distances = pdist(np.array(positions))
169     distance_matrix = squareform(distances)
170
171     hop_limit = system.hop_limit
172     for i in range(len(atoms)):
173         atom1 = atoms[i]
174         available = []
175         if atom1.charge > 0:
176             for j in range(len(atoms)):
177                 atom2 = atoms[j]
178                 if atom1.charge > atom2.charge:
179                     dist = distance_matrix[i][j]
180                     if dist < hop_limit and dist != 0:
181                         for k in range(int(atom2.Nel - atom2.charge
182 ))):
183                             available.append(j)
184                             if len(available) > 0:
185                                 target = np.random.choice(available)
186                                 atom1.charge -= 1
187                                 atoms[target].charge += 1
188
189     return atoms
190
191 # return modified Dee value
192 def get_charged_dee(atom1, atom2, Dee):
193     Nel1 = atom1.Nel
194     Nel2 = atom2.Nel
195     Q1 = atom1.charge
196     Q2 = atom2.charge
197     charged_De = Dee * ((Nel1 - Q1) + (Nel2 - Q2)) / (Nel1 + Nel2)
198     return charged_De
199
200
201 # add 1 charge to atom
202 def add_charge_to_atom(atoms, number, n_charge):
203     atoms[number].charge = n_charge
204     return atoms
205
206
207 # read atoms from input file

```

```

208 def read_input_file(filename, cons):
209     f = open(filename)
210     lines = f.readlines()
211     f.close()
212
213     f = open(cons)
214     c_lines = f.readlines()
215     f.close()
216
217     atoms = []
218     eq_positions = []
219     for line in lines[2:]:
220         parts = line.split(" ")
221         if len(parts) > 0:
222             type = parts[0]
223             positions = np.zeros((3))
224             positions[0] = float(parts[1])
225             positions[1] = float(parts[2])
226             positions[2] = float(parts[3])
227             positions *= A_to_m # A --> m
228             mass = float(parts[4]) * amu_to_kg # u --> kg
229             Nel = float(parts[5])
230             rad = float(parts[6]) * A_to_m
231             velocities = np.zeros((3))
232             atoms.append(Atom(type, positions, velocities, mass,
233                               Nel, rad))
234             eq_positions.append(positions)
235
236     for i in range(len(c_lines)):
237         connection = [int(x) for x in c_lines[i].split(',')]]
238         atoms[i].connections = connection
239
240     eq_distances = pdist(np.array(eq_positions))
241     eq_distances = squareform(eq_distances)
242     return atoms, eq_distances
243
244 # vector for force
245 def f_vector(atom_1, atom_2, dist):
246     fvec = np.zeros(3)
247     for i in range(3):
248         fvec[i] = (atom_2.position[i] - atom_1.position[i]) / dist
249     return fvec
250
251 # calculate system energy
252 def calc_energy(energy_file, atoms, system, eq_distances, round):
253     epstot = 0.0
254     epmtot = 0.0
255     ektot = 0.0
256
257     positions = []
258     for atom in atoms:
259         positions.append(atom.position)
260     distances = pdist(np.array(positions))
261     distance_matrix = squareform(distances)

```

```

262     for i in range(len(atoms)):
263         epm = 0.0
264         eps = 0.0
265         for j in range(len(atoms)):
266             if i == j:
267                 continue
268
269                 atom_i = atoms[i]
270                 atom_j = atoms[j]
271
272                 Re = eq_distances[i][j]
273                 x = distance_matrix[i][j]
274
275                 charge1 = atom_i.charge * e_to_C
276                 charge2 = atom_j.charge * e_to_C
277
278                 eps += k_e * (charge1 * charge2) / x
279
280                 if i not in atom_j.connections:
281                     continue
282
283                 Dee, Eq, m, k, spring_r = get_params(system, atom_i,
atom_j)
284                 b = Eq / 2 / hbar * np.sqrt(2 * m / Dee)
285
286                 epm += Dee * (1 - np.exp((-1) * b * (x - Re))) ** 2 -
Dee
287                 velocity = np.sqrt(atoms[i].velocity[0] ** 2 + atoms[i].
velocity[1] ** 2)
288                 ekin = 0.5 * atoms[i].mass * velocity ** 2
289
290                 ektot += ekin
291                 epstot += eps
292                 epmtot += epm
293
294                 line = str(round) + ',,' + str(ektot) + "," + str(epmtot / 2) +
",," + str(epstot) + "\n"
295
296                 energy_file.write(line)
297
298
299 # type, position, velocity, mass, Nel, rad, connections=0, charge=0
300 def write_system_output(output_file, atoms, round):
301     for atom in atoms:
302         line = str(round) + ',,'
303         line += atom.type + ',,'
304         line += str(atom.position[0]) + ',,'
305         line += str(atom.position[1]) + ',,'
306         line += str(atom.velocity[0]) + ',,'
307         line += str(atom.velocity[1]) + ',,'
308         line += str(atom.charge) + '\n'
309         output_file.write(line)
310
311
312 # write current atom positions to file

```



```

313 def write_trajectory(file, atoms):
314     file.write(str(len(atoms)) + "\n")
315     file.write("type x y \n")
316     for atom in atoms:
317         line = str(atom.type) + " "
318         line += str(atom.position[0] / A_to_m) + " "
319         line += str(atom.position[1] / A_to_m) + "\n"
320     file.write(line)
321
322 # randomize starting charge locations
323 def randomize_start(atoms, charges):
324     for q in range(charges):
325         selection = []
326         for i in range(len(atoms)):
327             atom = atoms[i]
328             for j in range(int(atom.Nel - atom.charge)):
329                 selection.append(i)
330         pick = np.random.choice(selection)
331         atoms[pick].charge += 1
332     return atoms
333
334
335 # make file command line run-able to enable multiple instances
336 # args: tag, gamma, iter_c, eint, identifier
337 # run example: python simulation_presentable.py dicat 1E15 5 10 i1
338 def main(args):
339
340     dt = 0.5E-17 # 1E-17
341     time = dt * 1000000
342
343     nametag = args[0]
344     gamma = float(args[1])
345     iter_c = int(args[2])
346     E_int = int(args[3])
347     identifier = args[4]
348
349     print('type: {}, gamma: {}, iteration count: {}, internal
350 energy: {}'.format(nametag, gamma, iter_c, E_int))
351
352     datetag = datetime.date.today().isoformat()
353
354     output_name = 'output_data\\' + nametag + '\\output_' + datetag
355     + '_' + nametag + '_' + str(E_int) + '_eint_' + identifier + '.
356 txt'
357     energy_name = 'output_data\\' + nametag + '\\energy_' + datetag
358     + '_' + nametag + '_' + str(E_int) + '_eint_' + identifier + '.
359 txt'
360
361     output_file = open(output_name, 'w')
362     energy_file = open(energy_name, 'w')
363
364     output_file.write('round,type,xpos,ypos,xvelocity,yvelocity,
365 charge\n')
366     energy_file.write('round,ekin,emorse,ecoulomb\n')

```

```

362
363     for i in range(iter_c):
364
365         trajectory_name = 'output_data\\' + nametag + '\\trajectory
\\trajectory_' + nametag + '_' + str(E_int) + '_eint_' +
identifier + '_' + str(i) + '.txt'
366
367         trajectory_file = open(trajectory_name, 'w')
368
369         system = System(hop_limit=2.5,
370                        hop_frequency=int(1 / (gamma * dt)),
371                        k=1000,
372                        internal_bond_energy=E_int / 9,
373                        frag_limiter=2.5)
374
375         atoms, eq_distances = read_input_file("input.txt", "
con_input.txt")
376
377         if nametag == 'cascade':
378             atoms = randomize_start(atoms, 4)
379         elif 'dicat' in nametag:
380             atoms = randomize_start(atoms, 2)
381         else:
382             print('nametag wrong')
383
384         calc_energy(energy_file, atoms, system, eq_distances, i)
385
386         atoms = leapfrog_integrator(atoms, system, eq_distances,
trajectory_file, dt, time, connection=1)
387
388         calc_energy(energy_file, atoms, system, eq_distances, i)
389
390         write_system_output(output_file, atoms, i)
391
392         print('{} / {} completed'.format(i + 1, iter_c))
393
394         trajectory_file.close()
395
396     output_file.close()
397     energy_file.close()
398
399
400 if __name__ == "__main__":
401     main(sys.argv[1:])

```