

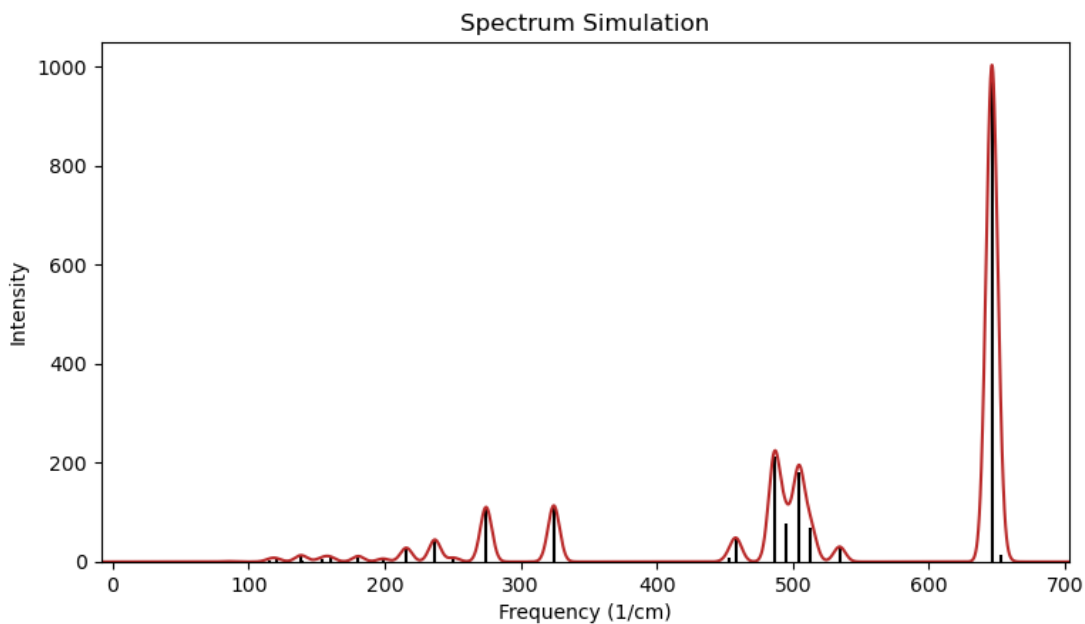
Projektityön raportti: Spektrisimulaatio

Johanna Hakonen

8.5.2020

Yleiskuvaus

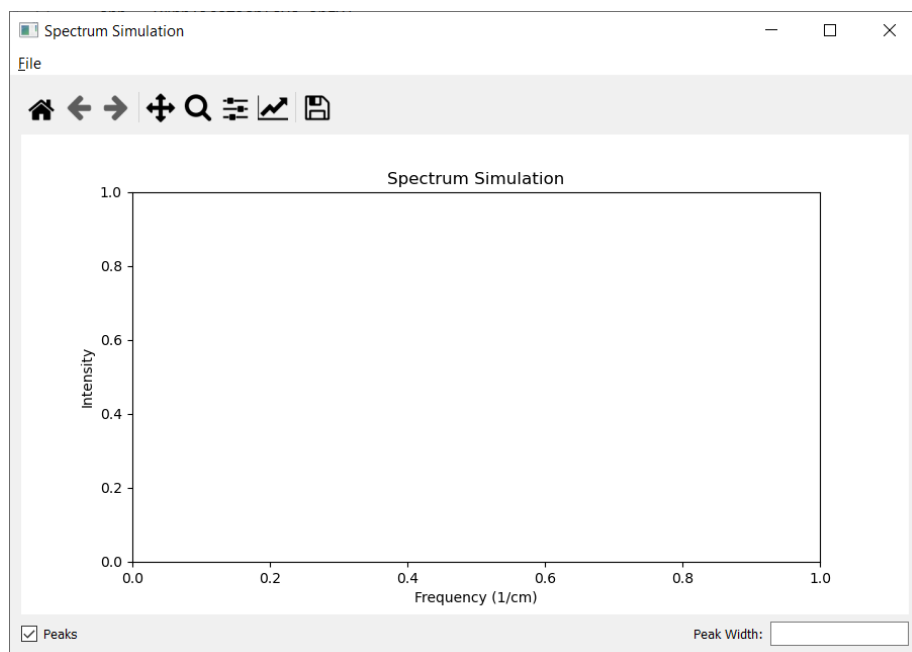
Projektityössä toteutettiin spektrisimulaatio-ohjelma, jolla piirretty ja tallennettu Raman-värentelyspektri on esitetty Kuvassa 1. Spektrin x-akselilla on värähdyksen energia (aallonpituus/taajuus cm^{-1}) ja y-akselilla intensiteetti. Ohjelma lukee käyttöliittymässä avatusta tekstitiedostosta molekyylin kvanttikemiallisesti lasketun spektrin sekä piirtää sen piikkeinä ja kokeellista mittausta simuloivana spektrinä, jossa piikit on levennetty. Kokeellisesti mitatut piikit ovat leveämpiä johtuen mittalaitteen tarkkuudesta ja näytteestä, jolloin vierekkäiset piikit eivät erotu toisistaan. Käyttäjä voi säätää käyttöliittymässä kuvaajan asetuksia, kuten otsikoita, akseleiden välejä, piikkien leveyttä ja sitä, näkykö piikit. Projektissa myös testattiin ohjelman toimivuutta. Työ oli tarkoitus toteuttaa keskivaikeana mutta valmiissa NavigationToolbar2QT-luokassa [1] oli valmiina mukana vaikean tason vaatimuksista spektrin tallennus kuvatiedostona, joten toteutus oli laajempi.



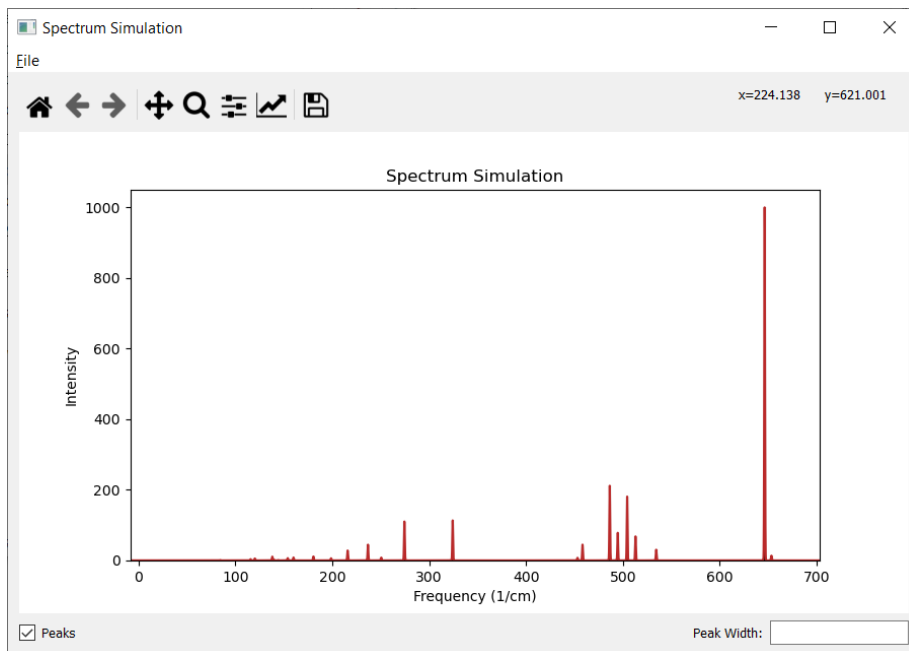
Kuva 1. Simulointiohjelmalla piirretty ja tallennettu laskennallinen (mustat piikit) ja simuloitu kokeellinen (punainen käyrä) spektri.

Käyttöohje

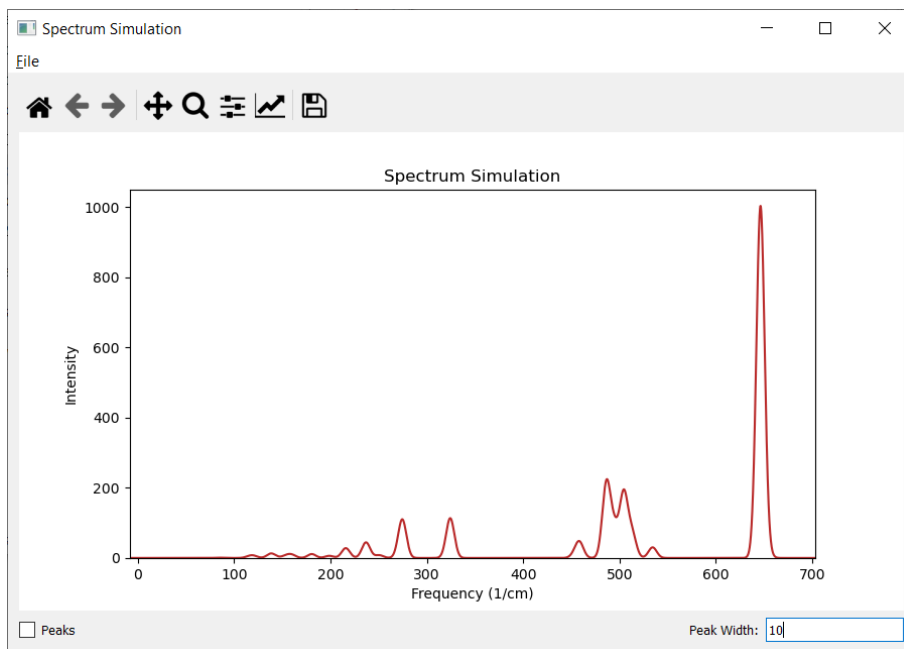
Ohjelma käynnistetään ajamalla pääohjelma. Tekstitiedosto avataan valikkopalkista (File – Open). Työkalupalkilla tehdään muutoksia kuvaajaan ja se näyttää kursorin paikan. Koti-napista palataan alkuperäiseen näkymään. Nuolella vasemmalle palataan edelliseen näkymään ja nuolella oikealle mennään seuraavaan näkymään. Nuolipäisellä ristillä siirretään akseleiden nollakohtia (hiiren vasen näppäin painettuna alas) tai zoomataan kuvaajaa (hiiren oikea näppäin). Suurennuslasilla zoomataan suorakulmioksi. Kuudennella säätönapilla muutetaan kuvan kokoa. Seitsemännellä muokataan kuvaajan otsikoita, akseleiden alku- ja loppuarvoja sekä viivan ominaisuuksia. Tallennusnapilla tuotetaan kuva png-muodossa. Laskennalliset piikit saa pois näkyvistä ”Peaks”-valintaruudun avulla. Piikkien leveyttä vaihdetaan ”Peak Width”-tekstikentällä. Käyttöliittymä on esitetty alkutilanteessa Kuvassa 2, latauksen jälkeen Kuvassa 3 sekä piikkien poiston ja levennyksen jälkeen Kuvassa 4.



Kuva 2. Käyttöliittymä alussa.



Kuva 3. Käyttöliittymä tiedoston avaamisen jälkeen.



Kuva 4. Käyttöliittymä piikkien poiston ja levennyksen jälkeen.

Ulkoiset kirjastot

Projektissa käytettiin Pythonin NumPy [2]- ja Matplotlib [3,4]-kirjastoja. NumPya tarvitaan taulukon luomiseen kuvaajaa varten ja Matplotlibia kuvaajan piirtämiseen. Käyttöliittymäkirjastona käytetään Qt:ä PyQt5:n [5] avulla.

Ohjelman rakenne

Pääohjelma luo Qt-sovelluksen vaatiman PyQt5:n QtWidgets-moduulin QApplication-ilmentymän [6] ja käynnistää graafisen käyttöliittymän. Käyttöliittymää kuvaa oma GUI-luokka, joka pohjautuu PyQt5:n QtWidgets-moduulin QMainWindow-luokkaan. Siinä käytetään PyQt5:n QtWidgets-moduulin luokkia. QMainWindow-luokka käyttää osien asettelua varten centralWidget-ilmentymää, johon asetetaan QWidget-luokan ilmentymä, joka on pohja muille käyttöliittymäolioille. CentralWidget:iin asetetaan pystysuuntainen QVBoxLayout-ilmentymä, johon asetetaan vaakasuuntaisia QHBoxLayout-ilmentymiä eri tehtäviä varten.

Kuva tehdään GUI:ssa omalla PlotFigure-luokalla, joka perii matplotlib.backends.backend_qt5agg-moduulin FigureCanvasQTAgn-luokan [1,7,8]. PlotFigure lukee tiedoston ja piirtää kuvan. Siinä käytetään valmista Pyplot-luokkaa [4,9] Matplotlib-kirjastosta kuvaajien muodostamiseen. Kuvaajan pohjakuva on Pyplot.figuren ilmentymä [10]. Virheiden käsittelyyn tehtiin myös oma CorruptedDataFileError-luokka.

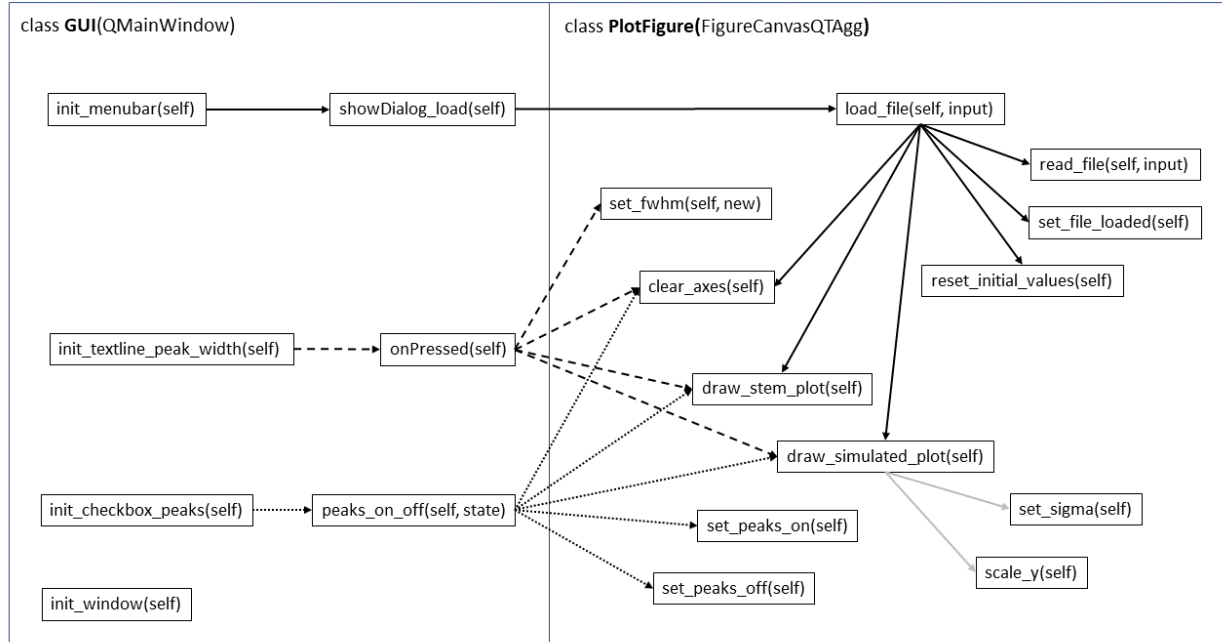
GUI- ja PlotFigure-luokkien metodit ja niiden suhteet on esitetty Kuvassa 5. Käyttöliittymän init_menubar-metodi alustaa alasvetovalikon. File-valikko muodostetaan QMenuBar-luokalla [11]. QAction-luokka liitetään File-valikkoon ja Open-tekstin kuva liitetään siihen PyQt5:n QtGui-moduulin QIcon-luokalla [6]. QAction kutsuu showDialog_load-metodia, kun klikataan File – Open. Se puolestaan avaa valitun tekstitiedoston ja välittää sen merkkijonona PlotFiguren load_file-metodille. Load_file-metodi kutsuu muita metodeita, jotta saadaan muodostettua kuva. Listat alustetaan reset_initial_values-metodilla ja kuva tyhjennetään clear_axes-metodilla. Read_file-metodi muuttaa tiedostosta saadun merkkijonon StringIO-luokan ilmentymäksi ja lukee siitä taajuudet ja intensiteetit listoihin. Draw_stem_plot-metodi piirtää listojen perusteella laskennalliset piikit [12,13] ja draw_simulated_plot-metodi simuloidun käyrän.

Käyttöliittymän työkalurivi kuvan säätöjä varten on NavigationToolBar2QT-luokan [1] ilmentymä Matplotlib.backends.backend_qt5-moduulista. Sille välitetään PlotFigure-luokan ilmentymä.

Käyttöliittymän init_checkbox_peaks-metodi alustaa valintaruudun piikkien näkyvyyden määrittämiseksi. Valintaruutu luodaan QCheckBox-luokalla [14], joka valintaa muutettaessa kutsuu peaks_on_off(state)-metodia. Tämä tarkistaa valintaruudun tilan PyQt5:n QtCore-moduulin Qt:n avulla. Kun valinta otetaan pois, kutsutaan PlotFiguren set_peaks_off-metodia, joka merkitsee piikit pois päältä. Lisäksi kutsutaan clear_axes-metodia ja draw_simulated_plot-metodia. Kun valinta laitetaan takaisin päälle, kutsutaan PlotFiguren set_peaks_on-metodia ja draw_stem_plot-metodia.

Käyttöliittymän piikinlevennyskenttä alustetaan init_textline_peak_width-metodilla. Peak Width-otsikko tehdään QLabel-luokalla ja tekstikenttä QLineEdit-luokalla [15]. Kun painetaan enteriä kentässä [16], se

kutsuu metodia onPressed, joka asettaa uuden piikin leveyden PlotFiguren set_fwhm(new)-metodilla, tyhjentää kuvan clear_axes-metodilla ja kutsuu piirtometodeita.



Kuva 5. Ohjelman ja käyttöliittymän rakenne.

Algoritmit

Simuloitujen yksittäisten piikkien ajatellaan noudattavan normaalijakauman tiheysfunktioita (Kaava 1) [17]. Sen ”keskihajonta”, σ , saadaan puoliarvoleveydestä (full width at half maximum, FWHM) Kaavoilla 2 ja 3. Odotusarvo x_0 vastaa tekstitiedoston taajuutta X . Yhden piikin simuloitu viivanmuoto y saadaan kaavasta 4 käyttämällä tekstitiedoston intensiteetin Y arvoja. Koko simuloitun spektrin yhtälö y_{sim} on kaikkien simuloitujen piikkien yhtälöiden summa (Kaava 5).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{(x-x_0)^2}{2\sigma^2} \quad (1)$$

$$FWHM = 2\sigma\sqrt{2\ln 2} \quad (2)$$

$$\sigma = \frac{FWHM}{2\sqrt{2\ln 2}} \quad (3)$$

$$y = Yf(x) \quad (4)$$

$$y_{sim} = \sum y \quad (5)$$

Lisäyksenä suunnitelmaan yksittäisten simuloitujen piikkien yhtälöt skaalataan, koska huomattiin, että säädettäessä puoliarvoveveyttä simuloitun piikin huipun y ei enää ollut sama kuin Y . Yhtälön $f(x)$ arvo huipussa ($x=X=x_0$) täytyy olla 1, jotta $y=Y$. Tällöin $f(x)$ saa Kaavassa 6 esitetyn muodon, jonka perusteella piikin yhtälö skaalataan tekijällä $FWHM/FWHM_0$. $FWHM$ on käyttäjän antama arvo ja $FWHM_0 \approx 0,94$ saadaan Kaavasta 6.

$$f(x) = \frac{2\sqrt{2 \ln 2}}{FWHM\sqrt{2\pi}} = 1 \quad (6)$$

Tietorakenteet

Uuden tiedoston taajuudet ja intensiteetit luetaan aina samoihin muuttujiin, joten niiden täytyy olla muuttuvia eli listoja. Lähtötiedot olisi voitu lukea myös sanakirjaksi mutta lista tuntui tutummalta. Kuvaajan piirtäminen vaatii x - ja y -pisteet Numpy-taulukon [18,19] muodossa.

Tiedostot

Ohjelmassa ladataan tiedot tekstitiedostosta, Kuvassa 6 on esitetty esimerkki SrU2F12_Raman.txt tehtävänannosta. #-Merkillä alkavat rivit ovat kommenttirivejä, taajuus (X) on rivillä ensimmäisenä ja intensiteetti (Y) toisena.

```

# SrU2F12 Raman spectrum
# X = Freq (cm^-1); Y = Polycrystalline Raman intensity (arb. units)
41.75      0.1
84.27      0.95
88.65      0.0
115.74     3.67
120.22     5.44
138.22     10.61
139.80     2.2
144.23     0.9
154.13     6.45
160.08     8.26
180.66     11.22
192.46     0.04
198.86     6.14
215.94     28.08
236.93     44.56
250.66     8.0
270.59     0.48
274.57     110.09
324.32     113.52
453.15     7.39
458.32     44.78
486.40     211.39
494.64     78.15
504.37     181.35
513.02     68.03
534.32     30.33
646.23     1000
653.24     13.5

```

Kuva 6. Esimerkkitekstitiedosto SrU2F12_Raman.txt.

Testaus

Ohjelman testausta suunniteltiin tehtävän ohjelmoinnin rinnalla eri osien valmistuessa mutta se jäi kuitenkin loppuun ja vähäiseksi. Suunnitelman mukaisesti testattiin, muodostaako `PlotFigure`-luokan `read_file`-metodi oikeat listat lähtötiedoista, jotka ovat virheettömiä, jotka ovat luettavissa mutta joissa on ylimääräisiä tyhjiä merkkejä ja virheellisiä rivejä, ja täysin virheellisistä tiedoista. Testauksessa käytettiin lähtötietoina tekstitiedoston sijaan merkkijonoa. Ohjelma läpäisee nämä testit.

Käyttäjän tekemien muutosten onnistumista ei ehditty testata. Syötteiden sopivuuden käsittelyä ei toteutettu eikä siten myöskään testattu. Laskennallisten piikkien poiston onnistumista ja piikin leveyden muutoksen onnistumista ei ehditty testata. Virheilmoituksia ei toteutettu.

Ohjelman tunnetut puutteet ja viat

Ohjelma kaatuu, jos yrittää avata väärää tiedostotyyppiä kuten Excel-tiedostoa. Tämän pystyy mahdollisesti korjaamaan `CorruptedDataFileError`-luokan ilmentymällä `except OSError`-kohdassa. Nyt se luodaan vain `read_file`ssä.

Ohjelma kaatuu, jos piikin leveyden säätökenttään laitetaan muuta kuin kokonaisluku tai desimaaliluku (pisteellä). Tekstikentän sallitun syötteen voisi määritellä `QValidator`-luokan avulla [20]. Kuvan voisi myös tehdä uudestaan vain, jos kenttään laitetaan eri luku kuin aiemmin eli tarkastamalla, muuttuiko luku `get_fwhm`-metodilla.

Piikin leveys alussa on Kaavasta 6 saatu 0,94, joka voisi olla isompi, jotta laskennalliset piikit näkyisivät paremmin. Alkuarvo ei näy piikin levennys –tekstikentässä, sen voisi saada näkyviin `setText`-komennolla [25].

Simuloidun spektrin skaalaus tehtiin koodissa `Y`:lle (`scaled_y`) mutta matemaattisesti se olisi kuulunut tehdä `f(x)`:lle ja onnistunut ilman `scale_y`-metodia ja `scaled_y`-listaa kertomalla simuloidun spektrin `sum_y` `FWHM/FWHM0`:llä ennen piirtämistä. Alussa oli vaikea hahmottaa yhtälön käsittelyä ohjelmoinnissa, joten skaalaus oli helpompi liittää `Y`:n arvoon.

Uutta tiedostoa ladattaessa vaihtuu käyttäjän vaihtamat otsikot takaisin alkuperäisiin ja se voi olla ärsyttävää, jos käyttäjä haluaa avata ja printata monta kuvaa samoilla omilla otsikoilla. Täytyisi joko tehdä omat tekstikentät otsikoiden syöttämiseen tai perehtyä `NavigationToolbar2QT`-luokkaan, miten käyttäjän otsikot saa sieltä talteen. Mahdollisesti se onnistuisi uuden tiedoston avaamisen yhteydessä `axes.get_ylabel` [21], `axes.get_xlabel` [22] ja `axes.get_title` [23]-komennolla.

GUI:n ja `PlotFigure`n rakenteet eivät välttämättä ole ideaalisia. Piikinlevennysmetodi voisi kutsua `PlotFigure`sta yhtä metodia, joka kokoaisi tehtävän kutsumalla muita metodeita kuten `load_file`. Myös `set_peaks_on` ja `set_peaks_off` voisivat olla kokoavia metodeita. `Set_file_loaded`-metodi on turha. `File_loaded`-muuttujan voisi vain asettaa `True`:ksi, kun tiedosto avattu. `PlotFigure`n `clear_axes(self)` saattaa olla turha, jos kuvan (Axes) pystyy tyhjentämään GUI:n puolella. Listojen sijaan olisi voinut olla järkevämpi käyttää sanakirjaa.

Sijoittelu voisi olla kauniimpi, en löytänyt vielä tekstikentän pituuden säätämiskeinoja, joten lisäsin valintaruudun ja tekstikentän väliin `addStretch(1)`-komennon [24]. Kuvaaja olisi paremman näköinen, jos akseleiden päätepisteet olisivat tasalukuja. Tämän voisi toteuttaa ehkä jakolaskun avulla sen sijaan, että lisätään 50 alkuun ja loppuun. Suunnitelman mukaiset virheilmoitukset voisi lisätä statuspalkin [11] avulla, esimerkiksi ilmoitus virheellisestä tiedostosta, jos piikkejä ei löydy ollenkaan.

Vahvuudet ja heikkoudet

Ohjelma on yksinkertainen mutta toimiva. Työkalurivin kanssa ohjelmasta tuli monipuolisempi. Olen ylpeä siitä, että ylipäänsä osasin ratkaista simuloidun piikin korkeuden skaalaamisen piikin leveyttä

säädettäessä. Hyvää on se, että käyttäjä voi valita piikin näkyvyyden ja leveyden ennen tiedoston lataamista eli ne eivät muutu aina avatessa uutta kuvaa.

Puutteita ja vikoja on pureskeltu ja korjausehdotuksia esitelty edellisessä kappaleessa, käyttäjän kannalta ikävimmät heikkoudet ovat ohjelman kaatuminen väärällä syötteellä piikin leveyttä säädettäessä ja jos vahingossa yrittää avata muunlaista tiedostoa kuin tekstitiedostoa. Käyttäjän vaihtamat otsikot vaihtuvat myös takaisin alkuperäisiin uutta tiedostoa ladattaessa.

Poikkeamat suunnitelmasta

Suunnitelmaan tuli paljon poikkeamia, koska etukäteen en osannut asioita ja opin käytännössä kokeilemalla. Arvioin kokonaisajaksi 80 tuntia kurssin tietojen perusteella. Aikaa kului huomattavasti enemmän koronaviruksen rajoitustoimien (kotona olevat etäkoululaiset) vuoksi, joten aikataulu venyi arvioidusta valmistumisviikosta 18 palautusviikolle 19. Toteutusjärjestys noudatti muuten suunnitelmaa mutta testaus jäi loppuun, koska en osannut alkaa testaamaan ennen toimivaa versiota. Testaus jäi myös vähäisemmäksi, koska sille jäi liian vähän aikaa.

Matematiikka oli haastavaa ja simuloidun spektrin skaalaus piikin levennyksessä tuli lisänä, kun huomasin, että piikin korkeudet muuttuvat samalla. Yhtälöiden muodostaminen ja käsittely koodissa oli suunnitelman jälkeen vielä epäselvää. Tietorakenteina käytettiin erillisiä listoja frequencies ja intensities niiden yhdistelmän sijaan.

Pääohjelmaan suunnitelluista tehtävistä alkuarvojen asetus siirtyi PlotFigureen, joka myös luo tyhjän kuvan alussa. Käyttöliittymässä Load-napin sijaan tekstitiedoston avaamiseen käytettiin valikkopalkkia [25]. Tarkoitus oli toteuttaa itse otsikoiden ja akseleiden välien muutosmahdollisuus tekstikentillä mutta ne löytyivät valmiina NavigationToolbar2QT-luokasta [1]. Siihen sisältyy myös tallennustyökalu, kuvaajan koon säätö, suurennuslasi, nuolet, joilla voi mennä eteenpäin ja taaksepäin muutoksissa ja paluu alkuperäiseen kuvaan.

Simuloidun spektrin muodostus onnistui yksinkertaisemmin kuin suunnitelmassa, luokkaa SimulatedSpectre ei tarvittu. Erillistä full_width_at_half_maximum-metodia ei tarvittu myöskään vaan se käsitellään muuttujana. Uusia metodeita olivat clear_axes, scale_y, load_file, set_file_loaded, set_peaks_on ja set_peaks_off. Joidenkin metodien nimet muuttuivat: calculated_peaks -> read_file, stem_plot -> draw_stem_plot ja simulated_plot -> draw_simulated_plot. Käyttöliittymän luokat ja metodit kehittyivät vasta suunnitelman palautuksen jälkeen.

Toteutunut työjärjestys ja aikataulu

Aikataulun suunnitelma ja toteutuma on kuvattu Taulukossa 1. Ohjasin kahden alakoululaisen etäkoulua samalla, joten oikeaa tuntimäärää on mahdotonta sanoa. Testaus jäi eniten jalkoihin ja tehtiin vasta lopussa. Suunnittelu taas jatkui rinnalla koko ajan, koska toteutus hahmottui vähitellen. Dokumenttia koostettiin myös alusta asti. Ohjelmoinnin eteneminen on kuvattu Taulukossa 2.

Taulukko 1. Projektin aikataulu.

| Osio | Viikot Suunnitelma | Viikot Toteutuneet | Aika (h) Suunnitelma | Aika (h) Toteutunut |
|--------------------------------|-----------------------|-----------------------|-------------------------|------------------------|
| Suunnittelu | 9-11 (18) | 9-19 | Yht. 35 | 46 |
| Yleis- ja tekninen suunnitelma | 9-10 | 9-10 | 28 | 28 |
| Suunnitelmademo | DL 11 | 11 | 0.3 | 1 |
| Lisäsuunnittelu | | 11-19 | 7 | 18 |
| Asennukset | - | 12 | - | 1 |
| Ohjelmointi | 11-18 (DL 19) | 12-19 | Yht. 45 | 96 |
| Ohjelma | 11-18 | 12, 14-19 | 32 | 63 |
| Testaus | 13-18 | 18-19 | 5 | 1.0 |
| Checkpoint | 13 (12-DL 14) | 14 | | |
| README | 14 | 14 | 1 | 1 |
| Organizing the repository | - | 14 | - | 1 |
| Projektityön dokumentti | 11-18 | 12, 14-19 | 5 | 30 |
| README (repositoryn sisältö) | 17 | 19 | 2 | 1 |
| Projektidemo | 18 (DL 20) | | 0.5 | |
| Yhteensä | | | 80 | 143 |

Taulukko 2. Työjärjestys.

| Pvm | main | GUI | | | | | | PlotFigure | | Testaus |
|---------------------------|------|-----------------|------------|-------------------------|------------------------|----------|---|------------|------------------------|---------|
| | | Kuvan upotus | Sijoittelu | Työ- kalu- palkki | Tiedoston avaaminen | Checkbox | Piikin- levennys- teksti- kenttä | Kuvaaja | Tiedoston lukeminen | |
| 11.3., 19.3., 30.3. | | | | | | | | | | |
| 1.4. | | | | | | | | | | |
| 4.4. | | | | | | | | | | |
| 11.4. | | | | | | | | | | |
| 14.4. | | | | | | | | | | |
| 17.4. | | | | | | | | | | |
| 18.4. | | | | | | | | | | |
| 19.4. | | | | | | | | | | |
| 20.4. | | | | | | | | | | |
| 21.4. | | | | | | | | | | |
| 22.4. | | | | | | | | | | |
| 23.4. | | | | | | | | | | |
| 24.4. | | | | | | | | | | |
| 25.4. | | | | | | | | | | |
| 26.4. | | | | | | | | | | |
| 1.5. | | | | | | | | | | |
| 2.5. | | | | | | | | | | |
| 4.5. | | | | | | | | | | |
| 5.5. | | | | | | | | | | |
| 6.5. | | | | | | | | | | |

Arvio lopputuloksesta

Ohjelma on yksinkertainen mutta toimiva ja siistinnäköinen. Se pystyy lukemaan virheellisenkin tekstitiedoston. Käyttäjä voi valita piikin näkyvyyden ja leveyden ennen tiedoston lataamista. Simuloidun piikin korkeus pysyy oikeassa kohdassa leveyttä muutettaessa. Valmiista työkaluriviluokasta löytyy myös kuvan tallennusmahdollisuus ja ylimääräisiä säätökeinoja.

Käyttäjän kannalta heikkoudet ovat ohjelman kaatuminen väärällä syötteellä piikin leveyttä säädettäessä ja jos vahingossa yrittää avata muunlaista tiedostoa kuin tekstitiedostoa. Käyttäjän vaihtamat otsikot vaihtuvat myös takaisin alkuperäisiin uutta tiedostoa ladattaessa. Simuloidun spektrin skaalaus tehtiin koodissa intensiteetille mutta matemaattisesti se olisi kuulunut tehdä viivanmuodolle $f(x)$. Muutenkin ratkaisumenetelmissä on varmasti parannettavaa. Testaus jäi hyvin vajaaksi mutta toisaalta ohjelmaa tuli testattua hyvin paljon käyttöliittymän kautta yrityksessäni saada eri osia toimimaan. Mikään huippusuoritus tämä ei prosessina ollut mutta koen oppineeni todella paljon. Parempaankin olisin pystynyt mutta koronakevät ja lasten etäkoulu vei ajan ja energian.

Luulen, että ohjelman rakenne sopii muutosten tai laajennusten tekemiseen, koska se on yksinkertainen. Käyttöliittymään on ainakin helppo lisäillä palikoita. Mielelläni harjoittelisin vielä erilaisten osien kuten liukusäätimen ja viivanmuodon valinnan lisäämistä ohjelmaan. Kemistinä olen analysoinut usean eri aineen seoksia kromatografialla, joten minua kiinnostaisi rakentaa eri aineiden massaspektrejä yhdistelevä ohjelma helpottamaan niiden tunnistamista. Haluaisin myös muuttaa ohjelman exe-muotoon.

-
- 1 Matplotlib Developers 2020, *matplotlib/lib/matplotlib/backends/backend_qt5.py*, Github, Inc., viewed 4 May 2020, <https://github.com/matplotlib/matplotlib/blob/a45c39940ca205facdcdbde860ea2fbd5d4ca687d/lib/matplotlib/backends/backend_qt5.py>.
 - 2 NumFOCUS 2020, *NumPy*, NumFOCUS, viewed 3 March 2020, <<https://numpy.org>>.
 - 3 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *matplotlib Version 3.1.3*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 3 March 2020, <<https://matplotlib.org>>.
 - 4 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *Usage*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 4 April 2020, <https://matplotlib.org/faq/usage_faq.html>.
 - 5 Bodnar, J. 2017, *Introduction to PyQt5*, Bodnar, J., viewed 11 April 2020, <<http://zetcode.com/gui/pyqt5/introduction/>>.
 - 6 Bodnar, J. 2017, *First programs in PyQt5*, Bodnar, J., viewed 11 April 2020, <<http://zetcode.com/gui/pyqt5/firstprograms/>>.
 - 7 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2017, *matplotlib.backends.backend_qt5agg*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 30 March 2020, <https://matplotlib.org/2.0.0/api/backend_qt5agg_api.html>.
 - 8 Forogh P. 2019, *How to Embed Matplotlib Graph in PyQt5*, Code Loop, viewed 30 March 2020, <<https://codeloop.org/how-to-embed-matplotlib-graph-in-pyqt5/>>.
 - 9 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *matplotlib.pyplot.ylim*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 30 March 2020, <https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.ylim.html>.
 - 10 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *matplotlib.backends.backend_qt5agg*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 30 March 2020, <https://matplotlib.org/3.2.0/api/_as_gen/matplotlib.pyplot.figure.html>.
 - 11 Bodnar, J. 2017, *Menus and toolbars in PyQt5*, Bodnar, J., viewed 11 April 2020, <<http://zetcode.com/gui/pyqt5/menustoolbars/>>.
 - 12 Pythontic.com 2020, *Stem Plot Using Python And Matplotlib*, Pythontic.com, viewed 4 April 2020, <<https://pythontic.com/visualization/charts/stem%20plot>>.
 - 13 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *Stem Plot*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 14 April 2020,

<https://matplotlib.org/3.2.1/gallery/lines_bars_and_markers/stem_plot.html#sphx-glr-gallery-lines-bars-and-markers-stem-plot-py>.

14 Bodnar, J. 2017, *PyQt5 widgets*, Bodnar, J., viewed 11 April 2020, <<http://zetcode.com/gui/pyqt5/widgets/>>.

15 The Qt Company Ltd. 2020, *QLineEdit Class*, The Qt Company Ltd, viewed 21 April 2020, <<https://doc.qt.io/qt-5/qlineedit.html>>.

16 Parwiz 2019, *PyQt5 Creating QLineEdit With returnPressed Signal*, Code Loop, viewed 21 April 2020, <<https://codeloop.org/pyqt5-creating-qlineedit-with-returnpressed-signal/>>.

17 Wikipedia 2019, *Full width at half maximum*, Wikimedia Foundation, Inc., viewed 3 March 2020, <https://en.wikipedia.org/wiki/Full_width_at_half_maximum>.

18 Educative, Inc. 2020, *How to convert a list to an array in Python*, Educative, Inc., viewed 14 April 2020, <<https://www.educative.io/edpresso/how-to-convert-a-list-to-an-array-in-python>>.

19 Hanxue 2018, TensorFlow: TypeError: only size-1 arrays can be converted to Python scalars, Flummoxed by IT, viewed 14 April 2020, <<http://flummox-engineering.blogspot.com/2018/09/tensorflow-keras-typeerror-only-size-1-arrays-can-be-converted-to-python.html>>.

20 The Qt Company Ltd. 2020, *QLineEdit Class*, The Qt Company Ltd., viewed 1 May 2020, <<https://doc.qt.io/qt-5/qlineedit.html#setValidator>>.

21 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *matplotlib.axes.Axes.get_ylabel*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 1 May 2020, <https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.axes.Axes.get_ylabel.html>.

22 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *matplotlib.axes.Axes.get_xlabel*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 7 May 2020, <https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.axes.Axes.get_xlabel.html>.

23 Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team 2020, *matplotlib.axes.Axes.get_title*, Hunter J., Dale D., Firing E., Droettboom M. and the Matplotlib development team, viewed 7 May 2020, <https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.axes.Axes.get_title.html>.

24 Bodnar, J. 2017, *Layout management in PyQt5*, Bodnar, J., viewed 11 April 2020, <<http://zetcode.com/gui/pyqt5/layout/>>.

25 Bodnar, J. 2017, *Dialogs in PyQt5*, Bodnar, J., viewed 11 April 2020, <<http://zetcode.com/gui/pyqt5/dialogs/>>.