

Projektityön tekninen suunnitelma: Spektrisimulaatio

Johanna Hakonen

48960U KeN

6.3.2020

Ohjelman rakennesuunnitelma

Ohjelman osat ovat tekstitiedoston tiedon lukeminen listaksi/taulukoksi (X, Y), simuloidun spektrin yhtälön ($y = \Sigma Yf(x)$) muodostaminen listan/taulukon avulla, kuvan piirtäminen laskennallisista piikeistä ja simuloidusta käyrästä sekä käyttöliittymä, joka näyttää kuvan ja jolla käyttäjä voi säätää sen esitystä.

Main-metodi alustaa asetusarvot, luo tyhjän kuvan, mahdollisesti jo kuvan piirtävän PlotFigure-olion, luo QApplication-instanssin ja käynnistää graafisen käyttöliittymän (luokka GUI). Käyttöliittymäluokassa GUI on metodit ikkunan, kuva-alueen, Load-napin, eri tekstikenttien ja Peaks-valintaruudun luontiin. Lisäksi siinä on ainakin metodi kuvan päivitykselle.

Tekstitiedosto luetaan PlotFigure-luokassa calculated_peaks-metodilla (se voisi olla myös oma luokkansa CalculatedPeaks), joka muodostaa listan/taulukon (X, Y), kun käyttöliittymässä painetaan Load-nappia. Kuvaaja muodostetaan PlotFigure-luokan stem_plot- ja simulated_plot-metodeilla. Kuvan skaalaa tai otsikoita muutetaan set_title/start/stop-metodeilla. Todennäköisesti PlotFigure-luokassa on myös get_title/start/stop-metodit.

Stem_plot-metodi käyttää luotua listaa/taulukkoa (X, Y) ja simulated_plot-metodi käyttää luokkaa SimulatedSpectre ($y = \Sigma Yf(x)$), jolle annetaan konstruktoriparametriksi lista/taulukko (X, Y). Luokassa SimulatedSpectre on metodi simulated_peaks, joka tekee listan piikkien $Yf(x)$ -yhtälöistä metodin peak_widening ($f(x)$) avulla. Tämän tarvitsema sigma saadaan sigma-metodista, joka puolestaan käyttää full_width_at_half_maximum-metodia, jossa on ainakin set_fwhm- ja get_fwhm-metodit. Yhtälöiden käsittelyä täytyy vielä selvittää, voiko niitä sisällyttää listaan ja täytyykö ne käsitellä metodin sijaan luokkina.

Käyttötapauskuvaus

Käyttäjä käynnistää ohjelman (kuvaajan paikalla ei ole mitään) ja lataa haluamansa tiedoston Load-napilla (luokka PlotFigure piirtää kuvaajan rakennesuunnitelman mukaisella tavalla). Aina käyttäjän tehdessä muutoksia käyttöliittymässä ohjelma piirtää uuden kuvan. Käyttäjä vaihtaa piikkien leveyttä "Peak Width"-tekstikentässä. Käyttäjä valitsee laskennalliset piikit pois näkyvistä ottamalla ruksin pois "Peaks"-valintaruudusta. Käyttäjä vaihtaa myös kuvaajan otsikon "Title"-tekstikentästä, x-akselin otsikon "X Title"-tekstikentästä ja y-akselin otsikon "Y Title"-tekstikentästä. Hän myös katsoo tarkemmin pienempää aluetta kasvattamalla x-akselin alkukohtaa otsikon "X Range" alta "Start:"-tekstikentässä ja pienentämällä loppukohtaa "Stop:"-tekstikentässä sekä tekemällä saman y-akselille otsikon "Y Range" alta.

Algoritmit

Käynnistettäessä ohjelma piirtää tyhjän kuvan ja Load-nappia painettaessa muodostetaan tekstitiedostosta lista/taulukko (X, Y). Kuvaaja piirretään ensin oletusasetuksilla (FWHM, otsikot, akselien alueet), joita käyttäjä voi sitten muuttaa. Matemaattiset kaavat on esitetty yleissuunnitelmassa yhtälöissä 1-4. Puoliarvoveyden FWHM avulla lasketaan sigma, jota tarvitaan X:n kanssa piikinmuodon $f(x)$ laskemiseen. Koko simuloitava spektri saadaan eri laskennallisten piikkien $Y_f(x)$:ien summana $\Sigma Y_f(x)$.

Tietorakenteet

Ohjelman tietorakenteita täytyy vielä pohtia. Tekstitiedoston jokainen piikki eli (X, Y)-rivi siirretään listaan tai taulukkoon. Jos tiedot ladataan aina samaan rakenteeseen, täytyy sen olla lista, jotta sitä voidaan muuttaa. Samoin on simuloitujen piikkien kanssa. Lisäksi en vielä löytänyt tietoa, pystyykö Matplotlibin avulla piirtämään kuvaajan suoraan funktion kaavalla vai täytyykö tehdä taulukko tai lista $\Sigma Y_f(x)$:n arvoista eri x:n arvoilla. Lisäksi epäselvää on, minkälaisia tietorakenteita yhtälöt vaativat.

Kirjastot

Tehtävänannossa sanotaan, että Pythonin NumPy- ja Matplotlib-kirjastojen käyttö on sallittua tässä tehtävässä. NumPya tarvitaan taulukon luomiseen/käsittelyyn ja Matplotlibia kuvaajan piirtämiseen. Käyttöliittymäkirjastona käytetään Qt:a PyQt5:n avulla.

Aikataulu

Projektityö pyritään tekemään viikkojen 11-18 aikana. Aikataulusuunnitelma on kuvattu taulukossa 1. Ensimmäisellä hahmotellaan Main-metodia, toteutetaan kuvaajien piirtäminen Matplotlibin avulla ja tehdään alustava versio käyttöliittymästä. Seuraavaksi muodostetaan lista tai taulukko tekstitiedostosta (NumPy:n avulla). Lopuksi käyttöliittymään ja PlotFigure-luokkaan lisätään säätömahdollisuudet käyttäjälle. Testausta tehdään rinnalla, kun on jotain koodia valmiina.

Taulukko 1. Projektityön aikataulu.

Osio	Viikot	Aika (h)
Suunnittelu	9-11 (18)	Yht. 35
Palautus	DL 10	
• Yleis- ja tekninen suunnitelma	9-10	27,5
Suunnitelmademo	DL 11	0,25
Ohjelmointi	11-18 (DL 19)	Yht. 45
Ohjelma	11-18	31,5
Testaus	13-18	5
Checkpoint	13 (12-DL 14)	
• README		1
Palautus	18 (DL 19)	
• Projektityön dokumentti	11-18	5
• Koodi		
• README (repositoryn sisältö)	17	2
Projektidemo	18 (DL 20)	0,5
Yhteensä		80

Yksikkötestaussuunnitelma

Ohjelman testausta tehdään ohjelmoinnin rinnalla eri osien valmistuessa. Ohjelman keskeisimpiä osia ovat tekstitiedoston lukeminen ja kuvaajien piirtäminen. Tiedoston lukevan `calculated_peaks`-metodin testauksessa tulisi käyttää tiedostoa, joka on virheetön, tiedostoja, jotka ovat luettavissa mutta joissa on ylimääräisiä tyhjiä merkkejä ja virheellisiä rivejä, ja täysin virheellisiä tiedostoja. Tulisi tarkistaa, vastaako valmiin listan/taulukon luvut tekstitiedoston lukuja. Kuvaajan piirtämisessä voisi ainakin testata, vastaako `simulated_peaks`-metodin tuottaman listan sisältö tekstitiedostoa ja vastaako `full_width_at_half_maximum`-metodin asettama FWHM käyttäjän antamaa. Virheilmoitukset tulisi käydä läpi.

Kirjallisuusviitteet ja linkit

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

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>>.

NumFOCUS 2020, *NumPy*, NumFOCUS, viewed 3 March 2020, <<https://numpy.org>>.

Forogh P. 2019, *How to Embed Matplotlib Graph in PyQt5*, Code Loop, viewed 3 March 2020, <<https://codeloop.org/how-to-embed-matplotlib-graph-in-pyqt5/>>.