

18.2.2015

Matriisin determinantin selvittäminen toimii nyt ja se on testattu ja dokumentoitu. Ongelmana on kuitenkin algoritmin hitaus yli 8-levyisillä neliömatriiseilla. Raakaan voimaan perustuva ratkaisu sisältää niin monta rekursiokutsua, ettei testikattavuus työkalu pysty generoimaan testikattavuusraporttia, jos testeissä on liian suuria matriiseja.

Lähden vääntämään vielä LU-hajotelmaa, jos vaikka saisin sen toimimaan.

Opiskelin LU-hajotelmaa Wikipediasta ja Cormenista ja ehkä hahmotin sen toimintaa selvemmin. Aiemmin en ole ymmärtänyt determinantin suhdetta hajotelmaan, enkä opiskelukiireiden painamana ole perehtynyt lineaarialgebaan syvemmin. Cormenissä esitellään LUP-hajotelmaa, joka vaikuttaa mielestäni monimutkaisemmalta. Wikipediassa on kuvailtu doolittle-algoritmia, jolle löysin ymmärrettävän pseudokoodin opiskeltavaksi.

19.2.2015

Toteutin LU-hajotelman matriisin determinantin selvittämiseksi. Kirjoitin uusille funktioille kasan testejä ja kommentit. Testikattavuus on riveille 99 % (tyhjää pääluokkaa ei ole testattu) ja 95 % muodoille (determinantti tiedostossa hidasta determinantti funktiota ei voi testata isommilla syötteillä). Toiminnallisuus on raakaan voimaan perustuvaa toteutusta huomattavasti nopeampi.

Aloitin seuraavaksi toteutus ja testausdokumenttien valmistelun sekä kirjoitan ohjelmalle käyttöliittymän. Toteutusdokumenttiin laittanen suorituskyykyvertailua, jota voinen tehdä ainakin matriisikertolaskun kahden eri toteutuksen ja determinantin kahden eri toteutuksen välillä. Testausdokumentissa kuvailen ohjelman testausta eli 350 riviä yksikkötestejä sekä pohdin niiden mahdollisia puutteita.

19.2.2015 Myöhemmin

Huomasin LU-hajotelman determinantti funktiossa jotain häikkää; tietyillä syötteillä se saa aikaiseksi divide-by-zero-poikkeuksen ja toisilla matriiseilla se laskee väärin. Debuggaan funktiota huomenna lisää ja koitan pusertaa sen kuntoon, niin pääsen viimeistelyn kimppuun.