Tietorakenteiden ja algoritmien harjoitustyön määrittelydokumentti

Risto Tuomainen

10. elokuuta 2014

Työn aihe

Harjoitustyöni tarkoituksena on toteuttaa ohjelma, joka suorittaa matriisilaskutoimituksia. Ohjelman tulee suorittaa peruslaskutoimitukset kuten matriisien sklalaarimonikerran, summan ja tulon laskeminen, sekä lisäksi laskea determinantti. Näiden ohella ohjelman tulee voida laskea matriisin käänteisalkio sekä determinantti ja suorittaa Gauss-Jordan eliminointi. Tietyt dekomponoinnit lienevät tarpeen perusoperaatioiden nopeaan suorittamiseen. Lisäksi ohjelmaa voi laajentaa esim. ominaisarvojen ja projektioiden laskemisen suuntaan. Käyttöliittymä tulee olemaan rehdisti tekstipohjainen. Oletusarvoisesti käyttäjä antaa matriisit listaamalla alkioita, mutta mahdollisesti ohjelma myös lukea esim. .csv-muotoisia taulukoita matriiseiksi.

Käytettävät tietorakenteet ja algoritmit

Harjoitustyössä on kaksi keskeistä teemaa matriisien tallentaminen tilaa säästävästi ja laskutoimitusten suorittaminen tehokkaasti. Matriiseja voi tallentaa järkevästi hieman eri tavoilla riippuen siitä millaisesta matriisista on kyse. Esim. diagonaalimatriisin voi tallentaa helposti pelkästään yksiulotteisena taulukkona, jossa on vain diagonaalin alkioiden määrän verran alkioita. Sen sijaan mikäli matriisissa on paljon nollia, sen tallentamiseen voi käyttää (muun muassa) compressed sparse column -formaattia, joka on myös tehokas peruslaskutoimitusten suorittamisessa. Matriisien tehokas tallentaminen riippuu kuitenkin melko paljon matriisin ominaispiirteistä ja käytettävistä algoritmeista. Matriisi, jossa jokaisessa alkiossa on jokin muu arvo kuin nolla, tarvitsee joka tapauksessa alkioidensa määrän verran muistipaikkoja.

Matriisin $A \in R^{n \times n}$ determinantin laskeminen naiivilla tavalla on aikavaatimukseltaan $O(n!)$. Tästä päästään aikavaatimukseen $O(n^3)$, mikäli ensin lasketaan $A = PLU$ dekomponointi, jonka determinantti saadaan helpommin selville.

Matriisien kertolasku on neliömatriisn tapauksessa $O(n^3)$. Harjoitustyössä toteutan Strassen algoritmin, jolla tästä päästään aikavaatimukseen $O(n^{2.8074})$. Tarjolla olisi toki muitakin algoritmeja, mutta Strassen algoritmi lienee sopivan simppeli, ollen kuitenkin naivia tapaa tehokkaampi. Matriisien kertolasku muuussa kuin neliömatriisien tapauksessa on toistaiseksi mysteeri.

Skalaarimonikerta $cA, c \in \mathbb{R}, A \in \mathbb{R}^{m \times n}$ on naivilla tavalla toteutettuna $O(n \times m)$. Tästä päästään aiemmin mainitun tallennusformaatin avulla aikavaatimukseen $O(z)$, missä $z = |\{x : x \in A, x \neq 0\}|$