Toteutusdokumentti

Ohjelman yleisrakenne

Ohjelmassa on tekstikäyttöliittymä, joka vastaanottaa käyttäjän syöttämät matriisit merkkijonona ja käyttäjän antamat laskuoperaatiot sekä muut käyskyt. Matriisivarasto muuntaa merkkijonon matriisiksi ja tallentaa sen varastoon. Ohjelmalla on rajapinta, joka määrittää matriisiolion pakolliset ominaisuudet, eli luokan Matriisi on palautettava matriisin sisältö, leveys ja pituus. Matriisin perusoperaatiot kutsutaan matriisioliolle ja niiden palautusarvo on matriisiolio. Matriisiluokasta löytyvät myös laskuoperaatioiden tarvitsemat apumetodit. Luokka LU on käytännössä laskin, joka suorittaa LU-hajotelman operaatiot parametriksi annetulle matriisioliolle ja myös palauttaa matriisiolion. Ne ovat omassa luokassaan siksi, koska kaikkia matriiseja ei voi laskea ylä- ja alamatriisin tulona. Ohjelma ei kuitenkaan erikseen kerro käyttäjälle, toteuttaako annetut matriisit LU -hajotelman ja determinantti lasketaan sen avulla kaikille matriiseille.

Ohjelman toiminnasta

Lu -luokan opiraatiot kutsutaan sellaisille matriiseille, jotka ovat jo hajautettuja doolittlella (paitsi doolittle itse, jota kutsutaan tavalliselle matriisille). Doolittlea kutsutaan matriisille siis käyttöliittymässä. Kertolaskussa käyttöliittymä kutsuu Strassen tuloa, joka puolestaan kutsuu naiivia tuloa jos matriisi tai sen alamatriisit ovat kooltaan noin alle 100x100 suuruiset. Päädyin tähän ratkaisuun, sillä empiirisen testauksen mukaan Strasse on tehokkaampi vasta sitä isommilla syötteillä. Doolittle ei toimi järin tehokkaasti isoilla syötteillä, mutta muilla operaatioilla syötteen koko ei jumita ohjelmaa niin pitkäksi aikaa. En ole varma laskeeko ohjelma oikean determinantin niille matriiseille joita ei voi hajottaa LU -matriiseiksi. Empiirisessä testauksessa nämä matriisit ovat palauttaneet oikean determinantin, mutta se voi olla sattumaa. Kuitenkin laskettaessa matriisin determinanttia operaatio toteutetaan kaikille LU -hajotelmalla.

Saavutetut aika- ja tilavaativuudet

Tilavaativuutena toimii matriisin koko. Naiivi matriisikertolasku toimii O(n^3) ajassa (kolme looppia ja loput toiminnot O(1) eli vakioajassa) ja Strassen matriisikertolasku tehokkaammassa O(n^2.8074) ajassa. Ohjelma käyttää Strassen tuloa ja siirtyy naiiviin algoritmiin kun matriisin koko on alle sadan. Metodit getL() ja getU() ovat O(n^2) (kaksi sisäkkäistä looppia) ja doolittle O(n^3) (korkeintaan kolme sisäkkäistä looppia). Käänteismatriisiin koodi on väärän tuloksen antavalla versiolla O(n^3) (korkeintaan kolme sisäkkäistä looppia). Tämä aikavaativuus on melko varmasti myös oikein toimivassa versiossa.

Suorituskyky- ja O-analyysivertailu

Empiirisen testauksen perusteella kertolaskualgoritmit noudattavat niitä O-vaativuuksia, mitä niiden oikein toteutettuna kuuluukin toteuttaa. Siis naiivi on O(n^3) ja Strasse O(n^2.8074). Kuten suorituskykytesteistä on käynyt ilmi, naiivi tapa on tehokkaampi pienillä ja Strasse isoilla syötteillä.

Työn mahdolliset puutteet ja parannusehdotukset

Koska työhön käytettävissä ollut aika oli rajallinen ja sairasteluiden sekä muiden kurssitöiden takia projekti eteni hieman aikataulustaan jäljessä, en onnistunut korjaamaan käänteismatriisioperaatiota toimivaksi. Metodi ei kaada ohjelmaa, mutta palauttaa väärät arvot puolelle alkioista. Tästä syystä päätin säilyttää koodin ohjelmassa, mutta vedin sen ja siihen liittyvät testit ja käyttöliittymän koodin harmaaksi. Ohjaaja voi halutessaan testata käänteismatriisin toimivuutta, mutta se on palautuksessani mukana epävirallisena osana. Myös testit etenkin LU-luokassa jäivät hieman vajaiksi, kun en ehtinyt niitä enempää kirjoittaa ja keskityin empiiriseen testaukseen ja suorituskykytesteihin. Jos jatkaisin työtä, toteuttaisin toimivan käänteismatriisin LU-hajotelman avulla ja myös perinteisellä tavalla, jolloin voisin vertailla näiden tapojen tehokkuutta.

Lähteet

LU-hajotelma: http://en.wikipedia.org/wiki/LU_decomposition
Doolittle: https://vismor.com/documents/network_analysis/matrix_algorithms/S4.SS2.php
Strassen algoritmi: http://en.wikipedia.org/wiki/Strassen_algorithm
Käänteismatriisi: http://www.ece.mcmaster.ca/~kiruba/3sk3/lecture6.pdf

Strassen pseudokoodi: http://www.google.fi/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CDMQFjAD&url=http%3A%2F%2Fhrcak.srce.hr%2Ffile%2F106985&ei=p8H8VO6KCcWzUeCigOAP&usg=AFQjCNHicoJUMH8uray9NP7mlzAkhytbqQ&sig2=uLAbl-ftKXbfsVK61oU0GA&bvm=bv.87611401,d.d24

(anteeksi hirveä linkki)