Matriisilaskimen määrittelydokumentti

- Mitä algoritmeja ja tietorakenteita toteutat työssäsi

  Harjoitustyössä on tarkoituksena käyttää Strassenin algoritmia matriisin kertolaskulle sekä LU-hajotelmaa matriisin determinantin laskemiseksi. Lisäominaisuuksiin voisi kuulua Gauss-Jordanin, käänteismatriisin sekä matriisin ominaisarvojen ja –vektoreiden laskemiseen käytettäviä algoritmeja tai hajotelmia.

- Mitä ongelmaa ratkaiset ja miksi valitsit kyseiset algoritmit/tietorakenteet

  Tavoitteena on ratkaista miten tehokkaasti laskea matriisin laskutoimituksia suuremmillakin syötteillä. Strassenin algoritmi on normaalia matriisin kertolaskua, joka on vaativuudeltaan $O(n^3)$, tehokkaampi algoritmi, jonka hyöty näkyy suurilla syötteillä. Vaikka on olemassa tehokkaampiakin algoritmeja, kuten Coppersmith-Winogradin algoritmi sekä vastoittain François Le Gallen optimoima algoritmi, ei Strassenista poikkeavia kuitenkaan käytetä käytännössä kuin harvoin.

  Strassenin algoritmi toimii rekursiivisesti hajota-ja-hallitse –periaatteella. Kerrottavista neliömatriiseista muodostetaan neljä samankokoista lohkomatriisia, jotka muodostavat osituksen matriisien ylä- ja alanurkiksi. Näiden sopivista summista ja erotuksista muodostetaan rekursiolla 7 eri tulomatriisia, joista lopuksi kootaan ratkaistava tulomatriisi. Koska normaalissa kertolaskualgoritmissa tulomatriisi voidaan rekursiivisesti muodostaa kahdeksalla kertolaskulla, säästämme Strassenilla ajallisesti yhden kertolaskun muokkaamalla laskettavia yhtälöitä sopivasti tyyliin ab + ac = a(b + c).

  Matriisin determinantin laskemiseksi päädyin käyttämään osittaistuettua LU-hajotelmaa, joka vaikuttaa QR-hajotelmaa sekä Choleskyn hajotelmaa yksinkertaisemmalta toteuttaa ja se toimii ilman lisärajoituksia reaaliarvoisille neliömatriiseille. Leibnizin ja Laplacen menetelmillä aikavaativuudeksi saadaan O(n!), kun taas LU-hajotelmalla päästään mukavampaan $O(n^3)$ vaativuuteen.

  LU-hajotelmassa neliömatriisista muodostetaan aluksi alakolmiomatriisi (L) ja yläkolmiomatriisi (U), joiden tulo LU esittää kyseistä neliömatriisia. Lisäksi vaaditaan, että alakolmiomatriisin lävistäjäalkiot ovat ykkösiä. Lopuksi on helppo laskea neliömatriisin determinantti yläkolmiomatriisin diagonaalialkioiden tulona. LU-hajotelmassa käytän lisäksi osittaistuentaa (partial pivoting), joka estää nollalla jakamisen ja parantaa numeerista stabiilisuutta.

- Mitä syötteitä ohjelma saa ja miten näitä käytetään

  Ohjelmalle voi manuaalisesti antaa matriiseja tai sitten ladata tiedostosta suuremman matriisin. Matriisit talletetaan 2-ulotteiseen array-taulukkoon, jonka alkiot ovat liukulukuja. Tämän jälkeen ohjelma laskee haluttavat laskutoimitukset ja ilmoittaa yleisiä tietoja matriisin ominaisuuksista.

- Tavoitteena olevat aika- ja tilavaativuudet (m.m. O-analyysit)

  Matriisin kertolasku: n x n matriiseille Strassenin algoritmin aikavaativuus $O(n^{\mathrm{lb}7}) = O(n^{2.8074})$ ja tilavaativuus $O(n^2)$

Matriisin determinantti: n x n matriiseille osittaistuetun LU-hajotelman aikavaativuus $O(n^3)$ ja tilavaativuus $O(n^2)$

- Lähteet

http://en.wikipedia.org/wiki/Strassen_algorithm

http://en.wikipedia.org/wiki/Coppersmith%E2%80%93Winograd_algorithm

http://en.wikipedia.org/wiki/LU_decomposition

http://en.wikipedia.org/wiki/QR_decomposition

http://en.wikipedia.org/wiki/Cholesky_decomposition

http://en.wikipedia.org/wiki/Determinant#Calculation

http://en.wikipedia.org/wiki/Pivot_element

Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms, 2nd edition