Tampereen yliopisto

Tommi Ilmonen

CHOLESKY-HAJOTELMA

Teknis-luonnontieteellinen
Kandidaatintyö
Huhtikuu 2019

TIIVISTELMÄ

Tommi Ilmonen: Cholesky-hajotelma
Kandidaatintyö
Tampereen yliopisto
TkK-tutkinto-ohjelma
Huhtikuu 2019

Tässä työssä esitellään Cholesky-hajotelma ja tarkastellaan hajotelman laskennallista tehokkuutta. Cholesky-hajotelmassa matriisi hajotetaan siten, että alkuperäinen hajotettu matriisi voidaan lausua hajotelmalla saadun yläkolmiomatriisin ja sen transpoosin tulona. Hajotelman avulla pystytään ratkaisemaan lineaarisia yhtälöryhmiä, joiden kerroinmatriisit ovat reaalisia, symmetrisiä ja positiivisesti definiittejä.

Työssä esitetään ensin tarvittavia määritelmiä ja lauseita Cholesky-hajotelma kannalta. Tämän jälkeen todistetaan hajotelman olemassaolo reaalisille, symmetrisille ja positiivisesti definiiteille matriiseille. Lisäksi esitellään perusteellisesti yksi tapa määrittää matriisille Cholesky-hajotelma. Tavalle johdetun algoritmin tehokkuutta sekä laskenta-aikoja tarkastellaan, ja saatuja tuloksia vertaillaan yleisesti tunnettuun LU-hajotelmaan. Vertailussa päädytään tulokseen, jossa Cholesky-hajotelma on tehokkuudeltaan optimaalisempi vaihtoehto. Lukijan oletetaan omaavan perustiedot ja käsitteet matriisilaskennasta.

Avainsanat: Cholesky-hajotelma, matriisihajotelma, laskennallinen tehokkuus

SISÄLLYSLUETTELO

1 JOHDANTO

Lineaarisia yhtälöryhmiä tulee vastaan hyvin usein eri tieteenaloilla sekä monissa erilaisissa ongelmissa. Lineaariset yhtälöryhmät voidaan esittää perinteisen yhtälöryhmän sijaan kerroinmatriisien ja vektoreiden tulona. Tällä tavalla tehtäessä yhtälöryhmä voidaan ratkaista kohdistamalla kerroinmatriisiin erilaisia operaatioita. Yleisesti ottaen yhtälöryhmien ratkaisemiseen on kehitetty useita eri tapoja ja algoritmeja, joissa ratkaisutavan lisäksi saattaa olla suuria eroja tehokkuudessa. Ratkaisualgoritmien tehokkuus yhtälöryhmien ratkaisussa on merkittävä tekijä tilanteissa, joissa ratkaistava yhtälöryhmä on hyvin suuri. Yksi tapa ratkaista lineaarisia yhtälöryhmiä on muodostaa yhtälöryhmää kuvaavalle kerroinmatriisille hajotelma, jonka avulla ratkaisu saadaan selville tehokkaasti. Eräs tällainen työkalu on Cholesky-hajotelma.

Tässä työssä keskitytään Cholesky-hajotelmaan, sen muodostamiseen matriisille sekä hajotelman tehokkuuteen. Cholesky-hajotelma on muodostettavissa kerroinmatriiseille, jotka ovat symmetrisiä ja positiivisesti definiittejä. Positiivisesti definiittejä ja symmetrisiä matriiseja esiintyy usein erilaisissa sovelluksissa ja niiden yhtälöryhmien kerroinmatriiseissa. Tälläisia kerroinmatriiseja tulee vastaan esimerkiksi sähköpiirien ja jousisysteemien matriisiyhtälöissä, sekä osittaisdifferentiaaliyhtälöiden numeerisissa ratkaisuissa. Työn tavoitteena on muodostaa käsitys siitä, kuinka Cholesky-hajotelma muodostetaan sekä millainen muodostusalgoritmin tehokkuus ja suorituskyky on.

Cholesky-hajotelmalla matriisi voidaan hajoittaa siten, että hajotettu matriisi voidaan lausua muodostetun yläkolmiomatriisin ja sen transpoosin tulona. Cholesky-hajotelma on yksi tehokkaimmista matriisihajotelmista eli sen muodostamiseen tietokoneella tarvitaan mahdollisimman vähän laskentaoperaatioita. Cholesky-hajotelma muistuttaa hyvin paljon LU-hajotelmaa ja usein ajatellaankin, että se on LU-hajotelman erityistapaus, jossa hajotettava kerroinmatriisi kattaa aikaisemmin mainitut välttämättömät ehdot. Tämän vuoksi LU-hajotelma on otettu mukaan työn vertailuosioon, jossa näiden kahden hajotelman suorituskykyjä ja toimintaa vertaillaan toisiinsa. Cholesky-hajotelma on mahdollista muodostaa sekä reaalisille että kompleksisille matriiseille. Tarkastelu tässä työssä on kuitenkin rajoitettu reaalisiin matriiseihin.

Työn alussa esitellään tarvittavia määritelmiä sekä lauseita Cholesky-hajotelman kannalta. Tämän jälkeen todistetaan hajotelman olemassaolo, sekä käydään perusteellisesti läpi algoritmi, jolla hajotelma saadaan muodostettua. Työn loppupuolella keskitytään kyseisen algoritmin tehokkuuteen, stabiilisuuteen sekä sen laskenta-aikoihin, ja vertaillaan algoritmin tehokkuutta LU-hajotelman vastaavaan. Näiden kahden hajotelman keskinäisessä vertailussa päästään tulokseen, jossa Cholesky-hajotelma on muodostettavissa noin puolet vähemmällä työllä kuin LU-hajotelma ja Cholesky on siten huomattavasti nopeampi vaihtoehto suuria matriiseja käsiteltäessä.

2 MATRIISILASKENNAN PERUSTEITA

Tässä luvussa esitetään työssä käytettyjä notaatioita sekä tarpeellisia määritelmiä ja lauseita Cholesky-hajotelman kannalta. Cholesky-hajotelma on olemassa vain symmetrisille ja positiivisesti definiiteille matriiseille, jolloin tarpeelliset käsitteet on hyvä ymmärtää. Seuraavaksi esitettävät määritelmät ja lauseet liittyvät positiivisesti definiittitteihin matriiseihin, sillä niitä tarvitaan tulevissa luvuissa käsiteltävissä aiheissa. Tämän luvun teoriaosuus myötäilee David S. Watkinsin kirjan [6] teoriaa samasta aiheesta. Osaa lauseista on hieman muutettu sekä niiden todistuksia avattu lukemisen helpottamiseksi.

2.1 Määritelmiä

Perinteisen matriisin esitystavan lisäksi matriisi voidaan esittää myös notaatiolla $\mathbf{A} = (a_{ij})$, missä alkion a indeksien i ja j rajat määräytyvät matriisin rivien i ja sarakkeiden j mukaan. Symmetrinen matriisi on määritelty matriisin *transpoosin* avulla. Matriisia transponoidessa sen sarakkeet ja rivit vaihtavat paikkaa keskenään eli matriisin (a_{ij}) transpoosi on (a_{ji}). Matriisin $\mathbf{A}$ transpoosille käytetään notaatiota $\mathbf{A}^{\mathbf{T}}$.

Määritelmä 2.1. Matriisi $\mathbf{A}$ on *symmetrinen*, jos se toteuttaa ehdon $\mathbf{A}^{\mathbf{T}} = \mathbf{A}$.

Symmetrinen matriisi on siis peilikuva diagonaalinsa suhteen eli toisin sanoen $(a_{ij}) = (a_{ji})$. Tämä tarkoittaa, että symmetrisen matriisin jokainen samalla indeksillä oleva rivi ja sarake ovat identtisiä. Alla esitettynä muutama esimerkki symmetrisistä matriiseista.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 4 & 5 \\ 1 & 4 & 1 & 1 \\ 3 & 5 & 1 & 2 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 3 & 1 & 2 & 7 & 0 \\ 1 & 2 & 4 & 1 & 0 \\ 2 & 4 & 5 & 3 & 1 \\ 7 & 1 & 3 & 6 & 2 \\ 0 & 0 & 1 & 2 & 2 \end{bmatrix} \tag{2.1}$$

Jos $n \times n$ matriisia kerrotaan oikealta puolelta n-alkioisella nollasta eroavalla sarakevektorilla $\mathbf{x}$ ja vasemmalta sen transpoosilla $\mathbf{x}^{\mathbf{T}}$, tuloksena on jokin skalaari $\mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x}$. Matriisin sanotaan olevan *positiivisesti definiitti*, jos syntyvä skalaari on aina positiivinen riippumatta vektorista, jolla sitä kerrotaan.

Määritelmä 2.2. Jos $n \times n$ neliömatriisi $\mathbf{A}$ on reaalinen, symmetrinen sekä toteuttaa ehdon $\mathbf{x^T A x} > 0$ kaikilla n-alkioisilla nollasta eroavilla vektoreilla $\mathbf{x}$, on matriisi *positiivisesti definiitti*.

Vaikka yllä olevassa määritelmässä positiivisesti definiitti matriisi määrättiin symmetriseksi, myös epäsymmetrinen matriisi voi toteuttaa ehdon $\mathbf{x^T A x} > \mathbf{0}$. Esimerkiksi matriisi

$$\begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix}$$

toteuttaa tämän ehdon kaikilla nollasta eroavilla vektoreilla $\mathbf{x}, \mathbf{y}$, sillä

$$\begin{aligned} &\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= 2x^2 + 2xy + 2y^2 \\ &= (x+y)^2 + x^2 + y^2 > 0. \end{aligned}$$

Määritelmän mukaiset, symmetriset ja positiivisesti definiitit, matriisit omaavat positiiviset ominaisarvot ja niiden myötä niiden determinantit ovat aina positiivisia. Koska näitä ominaisuuksia tullaan tarvitsemaan tulevissa luvuissa, on yllä oleva määritelmä on tässä työssä esitetty nimenomaan symmetrisille matriiseille.

Seuraavissa luvuissa tulee myös vastaan käsite *pääalimatriisi*. Matriisin pääalimatriisi saadaan ottamalla tietty määrä rivejä ja sarakkeita pois matriisin loppupäästä eli siis pohjasta ja oikeasta reunasta. Esimerkiksi edellisen sivun esimerkeistä (2.1) matriisin $\mathbf{B}$ kaikki pääalimatriisit ensimmäisestä viimeiseen ovat

$$\begin{bmatrix} 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 4 \\ 1 & 4 & 1 \end{bmatrix} \quad \text{ja} \quad \begin{bmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 4 & 5 \\ 1 & 4 & 1 & 1 \\ 3 & 5 & 1 & 2 \end{bmatrix}.$$

2.2 Lauseita

Puhuttaessa matriisin kääntyvyydestä tai ei-singulaarisuudesta, tarkoitetaan että matriisilla on olemassa käänteismatriisi eli inverssi, joka toteuttaa yhtälön $\mathbf{AA^{-1}} = \mathbf{I}$. Matriisin kääntyvyyden avulla saadaan useita tuloksia sekä tietoa sen käyttäytymisestä ja ominaisuuksista matriisiyhtälöissä. Alla olevassa lausessa esitetään matriisin kääntyvyydelle yhtäpitäviä ehtoja. Tämä tarkoittaa, että yhden ehdon ollessa voimassa ovat kaikki muutkin ehdot tosia.

Lause 2.1. *Olkoon $\mathbf{A}$ $n \times n$ neliömatriisi. Seuraavat neljä ehtoa ovat yhtäpitäviä.*

1. Matriisilla $\mathbf{A}$ on yksikäsitteinen käänteismatriisi $\mathbf{A}^{-1}$.

2. Yhtälöllä $\mathbf{Ay} = 0$ on vain triviaali ratkaisu $\mathbf{y} = 0$.

3. Yhtälöllä $\mathbf{Ax} = \mathbf{b}$ on yksikäsitteinen ratkaisu.

4. $\det(\mathbf{A}) \neq 0$.

Todistus. Lause 2.1 on todistettu lähes sellaisenaan kirjassa [1, s. 61 lause 4.3]. □

Tulevien lukujen kannalta lause on erittäin käytännöllinen, sillä lausetta voidaan soveltaa aina positiivisesti definiitille matriisille. Tämä todistetaan seuraavassa lauseessa. Lauseen todistuksessa osoitetaan, että positiivisesti definiitin matriisin determinantti on aina suurempi kuin nolla, jolloin yllä olevan lauseen kaikki ehdot toteutuvat.

Lause 2.2. *Lauseen 2.1 ehdot ovat voimassa positiivisisesti definiitille matriisille.*

Todistus. Positiivisesti definiitti matriisi toteutti määritelmänsä mukaan ehdon $\mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{x} > \mathbf{0}$ kaikilla ei-nollavektoreilla $\mathbf{x}$. Olkoon λ matriisin $\mathbf{A}$ jokin ominaisarvo ja $\mathbf{x}$ siihen liittyvä ominaisvektori. Tällöin matriisitulo $\mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{x}$ voidaan esittää matriisin $\mathbf{A}$ ominaisarvon avulla, jolloin $\mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{x} = \mathbf{x}^\mathbf{t}\lambda\mathbf{x}$. Ominaisarvon λ suhteen ratkaistuna saadaan

$$\lambda = \frac{\mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{x}}{\mathbf{x}^\mathbf{T}\mathbf{x}} = \frac{\mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{x}}{||\mathbf{x}||^2} > 0.$$

Positiivisesti definiitin matriisin ominaisarvot ovat yllä olevan yhtälön nojalla aina positiivisia. Matriisin $\mathbf{A}$ determinantti voidaan lausua sen ominaisarvojen tulona [2, s. 348]

$$\text{det}(\mathbf{A}) = \prod_{\mathbf{i}}^{\mathbf{n}} \lambda_\mathbf{i}.$$

Yhtölöstä nähdään, että positiivisesti definiitin matriisin determinantti positiivisten lukujen tulona on aina suurempaa kuin nolla ja lauseen 2.1 ehto 4. on voimassa. □

Joskus saattaa tulla vastaan tilanne, jossa voidaan muodostaa positiivisesti definiitti matrsiisi. Tälläinen tilanne esintyy muunmuassa pienimmän neliösumman menetelmissä, jossa matriisia kerrotaan sen transpoosilla. Seuraava lause antaa helpon tavan muodostaa positiivisesti definiittejä matriiseja kertomalla ei-singulaarinen matriisi transpoosillaan.

Lause 2.3. *Olkoon $\mathbf{M}$ jokin $n \times n$ ei-singulaarinen matriisi ja olkoon $\mathbf{A} = \mathbf{M}^\mathbf{T}\mathbf{M}$. Tällöin matriisi $\mathbf{A}$ on positiivisesti definiitti.*

Todistus. Osoitetaan ensin, että matriisitulosta $\mathbf{M}^\mathbf{T}\mathbf{M}$ syntyvä matriisi $\mathbf{A}$ on symmetrinen. Matriisitulon transpoosille on olemassa seuraavat laskusäännöt

$$(\mathbf{BC})^\mathbf{T} = \mathbf{C}^\mathbf{T}\mathbf{B}^\mathbf{T} \quad \text{ja} \quad \mathbf{B}^{\mathbf{TT}} = \mathbf{B}.$$

Näin ollen matriisin $\mathbf{A}$ transpoosi

$$\mathbf{A^T} = (\mathbf{M^T M})^\mathbf{T} = \mathbf{M^T M^{TT}} = \mathbf{M^T M} = \mathbf{A}.$$

Vielä pitää osoittaa, että matriisi $\mathbf{A}$ toteuttaa ehdon $\mathbf{x^T A x} > \mathbf{0}$ kaikilla nollasta eroavilla vektoreilla $\mathbf{x}$. Olkoon $\mathbf{y} = \mathbf{Mx}$ siten, että $\mathbf{y^T} = \mathbf{x^T M^T}$. Tällöin ehto $\mathbf{x^T A x} > \mathbf{0}$ saadaan muotoon

$$\mathbf{x^T A x} = \mathbf{x^T M^T M x} = \mathbf{y^T y} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{y_i^2} > 0.$$

Vektori $\mathbf{y}$ on ei-singulaarisen matriisin $\mathbf{M}$ ja nollasta eroavan vektorin $\mathbf{x}$ tulona aina erisuurta kuin nolla Lauseen 2.1 kohtien 2. ja 3. nojalla. Näin ollen $\mathbf{x^T A x} > \mathbf{0}$ kaikilla nollasta eroavilla vektoreilla $\mathbf{x}$ ja matriisi $\mathbf{A}$ on positiivisesti definiitti. $\square$

3 CHOLESKY-HAJOTELMA JA SEN LASKENTA

3.1 Cholesky-hajotelma

Cholesky-hajotelma on hyödyllinen työkalu symmetristen ja positiivisesti definiittien kerroinmatriisien hajottamiseen ylä- ja alakolmio muotoihin. Hajotelman avulla pystytään ratkaisemaan matriisiyhtälöitä, kunhan kerroinmatriisi kattaa Cholesky-hajotelman välttämättömät ehdot. Jos matriisille voidaan muodostaa Cholesky-hajotelma, myös LU-hajotelma kyseiselle matriisille on mahdollista muodostaa [2]. Cholesky-hajotelman etuna on kuitenkin, että sen algoritmi kolmiomatriisien muodostukseen on nopeampi kuin LU-hajotelman vastaava. Tätä käsitellään tarkemmin luvussa 4.

Lause 3.1. *Cholesky-hajotelma(Cholesky decomposition theorem). Olkoon $n \times n$ matriisi $\mathbf{A}$ posiitivisesti definiitti. Tällöin matriisi $\mathbf{A}$ voidaan muodostaa yksikäsitteisesti yläkolmiomatriisin $\mathbf{R}$ ja sen transpoosin $\mathbf{R^T}$ matriisitulona $\mathbf{A} = \mathbf{R^T R}$. Matriisia $\mathbf{R}$ kutsutaan matriisin $\mathbf{A}$ Cholesky-tekijäksi (Cholesky factor) ja sen diagonaalialkiot r_{ii} ovat positiivisia.*

Todistus. Todistus noudattelee N. J. Highamin kirjassa [3] esitettyä todistusta hajotelmalle. Todistetaan lause induktiolla. Lause on selvästi voimassa kun $n = 1$. Oletetaan, että lause on voimassa $n - 1$ kokoiselle matriisille. Tällöin oletuksen mukaan matriisin $\mathbf{A}$ pääalimatriisilla $\mathbf{A}_{n-1}$ on Cholesky-hajotelma $\mathbf{A}_{n-1} = \mathbf{R^T}_{n-1}\mathbf{R}_{n-1}$. Oletetaan seuraavaksi, että positiivisesti definiitille matriisille $\mathbf{A}$ voidaan muodostaa Cholesky-hajotelma $\mathbf{A} = \mathbf{R^T R}$ seuraavasti:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{n-1}} & \mathbf{c} \\ \mathbf{c}^T & a \end{bmatrix} = \mathbf{R^T R} = \begin{bmatrix} \mathbf{R}_{n-1}^T & \mathbf{0} \\ \mathbf{r}^T & b \end{bmatrix} \begin{bmatrix} \mathbf{R}_{n-1} & \mathbf{r} \\ \mathbf{0} & b \end{bmatrix} \tag{3.1}$$

Jotta ylläoleva hajotelma on mahdollista muodostaa, tulee löytyä yksikäsitteinen vektori $\mathbf{r}$ ja skalaari b, jotka täydentevät matriisit $\mathbf{R}$ ja $\mathbf{R^T}$ niiden pääalimatriisien kanssa. Matriisituloa (3.1) tarkastelemalla saadaan kyseiselle vektorille ja skalaarille yhtälöt

$$\mathbf{R}_{n-1}^T \mathbf{r} = \mathbf{c}, \tag{3.2}$$

$$\mathbf{r}^T \mathbf{r} + b^2 = a. \tag{3.3}$$

Koska matriisi $\mathbf{R}_{n-1}^T$ on kolmiomatriisi, sen determinantti on suurempaa kuin nolla positiivisten diagonaalialkioiden tulona. Täten voidaan soveltaa Lausetta 2.1 ja sen kohtaa 3.

Näin ollen yhtälöllä (3.2) on yksikäsitteinen ratkaisu vektorille $\mathbf{r}$. Yhtälöstä (3.3) voidaan ratkaista skaalari b, jolloin $b = \sqrt{a - \mathbf{r^T r}}$. Termin $a - \mathbf{r^T r}$ tulee olla suurempaa kuin nolla, jotta matriisi $\mathbf{R}$ pysyy reaalisena. Tämä voidaan osoittaa blokkimatriisin determinantti kaavalla

$$\det \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = \det(\mathbf{A} - \mathbf{B}\mathbf{D^{-1}}\mathbf{C})\det(\mathbf{D}) \quad [\mathbf{5}]. \tag{3.4}$$

Sovellettaan blokkimatriisin determinantti kaavaa kohdan (3.1) positiivisesti definiitille matriisille $\mathbf{A}$. Tällöin saadaan

$$\begin{aligned} \det(\mathbf{A}) &= \det \begin{pmatrix} \mathbf{A_{n-1}} & \mathbf{c} \\ \mathbf{c}^T & a \end{pmatrix} \\ &= \det(\mathbf{A}_{n-1} - \mathbf{c}\,\frac{1}{a}\,\mathbf{c}^T)\,a > 0. \end{aligned}$$

Oletuksen mukaan matriisi $\mathbf{A}_{n-1}$ voidaan esittää matriisien $\mathbf{R}_{n-1}^T$ ja $\mathbf{R}_{n-1}$ tulona. Vektori $\mathbf{c}$ saadaan yhtälöstä (3.2) ja sen transpoosi transponoimalla kyseinen yhtälö, jolloin

$$\begin{aligned} \det(\mathbf{A}_{n-1} - \mathbf{c}\,\frac{1}{a}\,\mathbf{c}^T)\,a &= \det(\mathbf{R}_{n-1}^T\mathbf{R}_{n-1} - \mathbf{R}_{n-1}^T\mathbf{r}\mathbf{r}^T\mathbf{R}_{n-1}\frac{1}{a})a \\ &= \det\left(\mathbf{R}_{n-1}^T(\mathbf{I} - \frac{1}{a}\mathbf{r}\mathbf{r}^T)\mathbf{R}_{n-1}\right)a \\ &= \det(\mathbf{R}_{n-1}^T)\det(\mathbf{R}_{n-1})\det(\mathbf{I} - \frac{1}{a}\mathbf{r}\mathbf{r}^T)a \\ &= \det(\mathbf{A}_{n-1})\det(1 - \frac{1}{a}\mathbf{r}^T\mathbf{r})a \\ &= \det(\mathbf{A}_{n-1})(a - \mathbf{r}^T\mathbf{r}) > 0. \end{aligned} \tag{3.5}$$

Yhtälöstä (3.5) on suoraan nähtävissä, että termi $a - \mathbf{r}^T\mathbf{r} > 0$, sillä matriisi $\mathbf{A}_{n-1}$ on positiivisesti definiitti. Näin ollen yksikäsitteinen Cholesky-hajotelma on olemassa. □

3.2 Cholesky-metodi ja algoritmi

Cholesky hajotelman laskemiselle on kehitetty useita eri algoritmeja. Tässä aliluvussa käsitellään näistä yhtä keskeisintä ja ehkä helposti ymmärrettävintä, sekä osoitetaan kuinka matriisi $\mathbf{R}$ saadaan muodostettua. Esiteltävässä algoritmissa hajotelma muodostetaan yksittäisten alkioiden ja niistä syntyvien sisätulojen avulla.

Toinen yleisesti käytetty tapa on määrittää hajotelma matriisiblokkien avulla. Tässä tavassa algoritmi toimii rekursiivisesti eli itse hajotelman muodostamiseen käytetään Cholesky-hajotelmaa pienemmissä osissa, jolloin suurten matriisien käsittelyssä tietokone pystyy käyttämään välimuistia ja rinnakkaisuutta erittäin tehokkaasti [6]. Tämän aliluvun sisältö on David S. Watkinsin kirjasta [6], mikäli toisin ei mainita.

Pyritään muodostamaan menetelmä, jolla Cholesky-hajotelma on mahdollista muodostaa. Kirjoitetaan matriisitulo $\mathbf{A} = \mathbf{R^T R}$ auki ja tarkastellaan sitä.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} r_{11} & 0 & 0 & \cdots & 0 \\ r_{12} & r_{22} & 0 & \cdots & 0 \\ r_{13} & r_{23} & r_{33} & & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ r_{1n} & r_{2n} & r_{3n} & \cdots & r_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & & r_{nn} \end{bmatrix}. \tag{3.6}$$

Matriisin $\mathbf{A}$ alkio a_{ij} on matriisin $\mathbf{R^T}$ i:nnen rivin tulo matriisin $\mathbf{R}$ j:nnen sarakkeen kanssa. Tarkastellaan matriisin $\mathbf{R^T}$ ensimmäistä riviä ja sen tuloa j:nnen sarakkeen kanssa, jolloin saadaan yhtälö

$$a_{1j} = r_{11}r_{1j} + 0r_{2j} + 0r_{3j} + \cdots + 0r_{nj} = r_{11}r_{1j}. \tag{3.7}$$

Alkion r_{11} määrittäminen on helppoa. Kun indeksi $j = 1$, saadaan yhtälöstä (3.7) $a_{11} = r_{11}^2$, josta saadaan ratkaistua matriisin $\mathbf{R}$ ensimmäinen diagonaalialkio $r_{11} = \pm\sqrt{a_{11}}$. Diagonaalialkioiden r_{ii} on oltava määritelmänsä mukaan positiivisia, eli valitaan positiivinen vaihtoehto $r_{11} = \sqrt{a_{11}}$. Kun on tiedossa ensimmäinen diagonaalialkio r_{11}, voidaan ratkaista matriisin $\mathbf{R}$ ensimmäisen rivin loput alkiot r_{1j} yhtälöstä (3.7).

$$r_{1j} = \frac{a_{1j}}{r_{11}}, \qquad j = 2, \ldots, n. \tag{3.8}$$

Saatu ensimmäinen rivi on samalla matriisin $\mathbf{R^T}$ ensimmäinen sarake. Matriisin $\mathbf{R^T}$ toista riviä määritettäessä ainoastaan kaksi alkiota r_{12} ja r_{22} ovat nollasta eroavia. Tällöin saadaan matriisitulosta (3.6) seuraava yhtälö samalla tyylillä kuin yhtälö (3.7), eli

$$a_{2j} = r_{12}r_{1j} + r_{22}r_{2j}. \tag{3.9}$$

Kun $j = 2$ yhtälö saa muodon $a_{22} = r_{12}^2 + r_{22}^2$. Tiedossa on jo alkion r_{12} arvo, jolloin yhtälöstä voidaan ratkaista toinen diagonaalialkio $r_{22} = \pm\sqrt{a_{22} - r_{12}^2}$. Tässäkin diagonaalialkion tapauksessa valitaan positiivinen vaihtoehto eli $r_{22} = \sqrt{a_{22} - r_{12}^2}$. Kun toinenkin diagonaalialkio on selvitetty, yhtälössä (3.9) ainoa tuntematon on alkio r_{2j}. Näin ollen voidaan määrittää matriisin $\mathbf{R}$ toisen rivin loput alkiot kyseisestä yhtälöstä ratkaisemalla se alkion r_{2j} suhteen, jolloin

$$r_{2j} = \frac{a_{2j} - r_{12}r_{1j}}{r_{22}}, \qquad j = 3, \ldots, n. \tag{3.10}$$

Käytetty menetelmä voidaan yleistää i:nnelle riville, olettaen, että $i-1$ edellistä riviä on määritetty samaan tapaan. Ainoastaan i ensimmäistä alkiota eroavat nollasta matriisissa $\mathbf{R^T}$ tarkasteltaessa i:nnettä riviä. Kerrottaessa matriisin $\mathbf{R}$ j:nnettä saraketta matriisin $\mathbf{R^T}$ i:nnellä rivillä saadaan yhtälö

$$a_{ij} = r_{1i}r_{1j} + r_{2i}r_{2j} + \ldots + r_{i-1,i}r_{i-1,j} + r_{ii}r_{ij}. \tag{3.11}$$

Koska ensimmäisten $i-1$:n rivin kaikki alkiot on jo määritetty, ainoat tuntemattomat alkiot ovat r_{ii} ja r_{ij}. Määritetään ensin diagonaalialkio r_{ii} asettamalla $j = i$. Tällöin yhtälö (3.11) saa muodon

$$a_{ii} = r_{1i}^2 + r_{2i}^2 + \cdots + r_{i-1,i}^2 + r_{ii}^2.$$

Josta edelleen saadaan ratkaistua tuntematon diagonaalialkio

$$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}. \tag{3.12}$$

Tämän jälkeen matriisin $\mathbf{R^T}$ määrittämättömän rivin i alkiot r_{ij} voidaan ratkaista yhtälöstä (3.11).

$$r_{ij} = \frac{\left(a_{ij} - \sum_{k=1}^{i-1} r_{ki}r_{kj}\right)}{r_{ii}}, \qquad j = i+1, \ldots, n. \tag{3.13}$$

Kyseistä menetelmää kutsutaan Cholesky-metodiksi (Cholesky's method). Menetelmään perustuvia tapoja määrittää Cholesky-hajotelma matriisille kutsutaan sisätuloalgoritmeiksi (inner-product formulation) yhtälöissä (3.12) ja (3.13) esiintyvien sisätulojen vuoksi.

Toisaalta yhtälöä (3.12) katsoessa voisi olettaa, että joillakin matriiseilla neliöjuuren sisällä oleva lauseke saisi joissakin tapauksissa negatiivisia arvoja, jolloin diagonaalialkiota r_{ii} ei voisi määrittää. Näin ei kuitenkaan positiivisesti definiitin matriisin tapauksessa voi käydä. Jos tälläinen tilanne tulee vastaan niin matriisi $\mathbf{A}$ ei ole positiivisesti definiitti. Cholesky-hajotelman muodostaminen toimiikin testinä matriisin positiivisesti definiittisyydelle ja sitä siihen käytetäänkin. Kaikki positiivisesti definiitit matriisit läpäisevät Cholesky-metodin ilman virheitä ja ei positiivisesti definiitin matriisin tapauksessa hajotelman muodostaminen pysähtyy jossain vaiheessa. Pysähtyminen tapahtuu juuri neliöjuuren sisällön mennessä negativiiseksi. [2, s. 164]

Eräs tälläinen sisätuloihin perustuva algoritmi on Cholesky-algoritmi (Cholesky's algorithm), jonka suoritusvaiheet perustuvat yllä esitettyyn Cholesky-metodiin. Alla olevan pseudokoodin tarkoitus on havainnollistaa algoritmin toimintaa yllä olevan avauksen lisäksi. Sanallisesti kuvattuna algoritmissa käydään silmukassa matriisin $\mathbf{A}$ jokainen rivi läpi. Jokaisen rivikäsittelyn aikana lasketaan ensin $\mathbf{R}$ matriisin r_{ii} diagonaalialkio k-silmukassa ja sen jälkeen r_{ii} diagonaalialkion oikeallapuolella samalla rivillä olevat alkiot r_{ij} j-silmukassa.

for *i* = *1*,. . . ,*n* **do**
  for *k* = *1*,. . . ,$i-1$ **do**
    $a_{ii} \leftarrow a_{ii} - a_{ki}^2$
  end
  $a_{ii} \leftarrow \sqrt{a_{ii}}$ (Tämä on r_{ii})
  for *j* = $i+1$,. . . ,n **do**
    for *k* = *1*,. . . , $i-1$ **do**
      $a_{ij} \leftarrow a_{ij} - a_{ki}a_{kj}$
    end
    $a_{ij} \leftarrow a_{ij}/a_{ii}$ (Tämä on r_{ij})
  end
end

Algoritmi 1: Esimerkki Cholesky-algoritmista [6].

Algoritmissa määritetään vain yläkolmiomatriisin diagonaali sekä sen yläpuolella olevat alkiot, sillä diagonaalin alapuolella alkiot ovat pelkkiä nollia eikä niitä luonnollisesti tarvitse laskea tai tallentaa mihinkään. Yläkolmiomatriisi voidaan tallentaa matriisin $\mathbf{A}$ alkioiden päälle, kuten pseudokoodissa tehdään, jolloin ylimääräistä tilaa ei tarvitse varata, sillä matriisin $\mathbf{R}$ alkioiden ollessa tiedossa ei matriisin $\mathbf{A}$ alkioita enään tarvita.

3.3 Laskenta-ajoista

Tarkastellaan Choleskyn-metodiin perustuvaa Cholesky-algoritmia ja sen vaatimaa laskentamäärää. Tietotekniikassa tietokoneen suorituskykyä mitataan flopseina (Floating point operations per second), joka siis kertoo kuinka monta laskutoimitusta tietokone pystyy suorittamaan sekunnin aikana. Algoritmien tehokkuudesta puhuttaessa tärkeää ei ole kauanko algoritmin määrittäminen kestää ajallisesti, sillä siihen vaikuttaa moni asia. Vaikuttavia asioita ovat muunmuassa tietokone, jolla algoritmi pyörii, ohjelmointikieli ja kääntäjät. Mielekkäämpää on puhua algoritmin käyttämien laskutoimitusten määrän muutoksesta syötteen koon muuttuessa. Eli matriisien tapauksessa voidaan puhua siitä miten algoritmin laskentaoperaatioiden määrä muuttuu, kun esimerkiksi algoritmiin syötettävän matriisin koko kaksinkertaistuu. Kun tarkastellaan algoritmin käyttämiä laskentaoperaatioita suoritusajan sijasta, voidaan eliminoida algoritmin suoritusaikaan vaikuttavat muuttujat pois tarkastelusta. Tällöin saadaan tarkasteltua algoritmin tehokkuutta olosuhteista riippumattomana kokonaisuutena, jossa kiinnitetään huomiota vain laskentaoperaatioiden määrään.

Cholesky-algoritmissa molemmissa k-silmukoissa tehdään kaksi laskutoimitusta. For-silmukoiden toistomäärä algoritmissa nähdään niille asetetuista rajoista. Tällöin i-silmukan sisällä olevassa ensimmäisessä k-silmukassa suoritetuille flopseille saadaan summaksi

$$\begin{aligned}\sum_{i=1}^{n}\sum_{k=1}^{i-1} 2 &= 2\sum_{i=1}^{n}(i-1) \\ &= n(n-1) \approx n^2. \end{aligned} \tag{3.14}$$

Samaan tapaan lasketaan toiseen k-silmukkaan käytetyt flopsit, jolloin kyseisen silmukan kokonaisflopsi määrä on

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=1}^{i-1}2.$$

Puretaan summa auki käyttämällä summakaavoja

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \quad \text{ja}$$
$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$$

Tällöin kokonaisflopsi määräksi saadaan

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=1}^{i-1}2 &= 2\sum_{i=1}^{n}\sum_{j=i+1}^{n}(i-1) \\
&= 2\sum_{i=1}^{n}(n-i)(i-1) \\
&= 2\sum_{i=1}^{n}(ni-i^2-n+i) \\
&= n^2(n+1)-\frac{2}{6}n(n+1)(2n+1)-2n^2+n(n+1) \\
&= \frac{n^3}{3}-n^2+\frac{2n}{3} \\
&\approx \frac{n^3}{3}. \qquad (3.15)
\end{aligned}$$

Huomiotavaa käytettyjen flopsien laskennassa on, että matriisin rivien määrän n kasvaessa suureksi, termi n^3 on huomattavasti suurempi kuin n^2. Näin ollen vaadittuja laskutoimituksia voidaan hyvin approksimoida jättämällä pienimmät potenssit tuloksesta pois. Ensimmäisen silmukan (3.14) flopsi määrä on huomattavasti pienempi isoilla n:n arvoilla kuin jälkimmäisen (3.15) silmukan. Täten Cholesky-algoritmin vaatiman flopsimäärän voidaan approksimoida olevan $n^3/3$ [6]. Kyseisellä algoritmilla matriisin koon kaksinkertaistuessa algoritmin vaatima flopsimäärä siis kahdeksan kertaistuu.

3.4 Numeerinen stabiilius

Numeriisella stabiiliudella tarkoitetaan algoritmin suorituksen tasapainoisuutta. Tietokone pyöristää liukulukuja laskutoimituksissa ja moneen kertaan tapahtuvat pienet virheet voivat muodostaa suurta virhettä algoritmin lopputulokseen. Algoritmin sanotaan olevan numeerisesti stabiili, jos algoritmin suorituksessa virheet eivät moninkertaistu. Tällöin suorituksen aikaiset virheet pysyvät vaaditun tarkkuuden sisällä ja saatua tulosta voidaan pitää luotettavana.

Tarkastellaan Cholesky-hajotelman stabiiliutta kahden tuloksen kautta. Jos matriisiyhtälölle $\mathbf{Ax} = \mathbf{b}$ muodostetaan Cholesky-hajotelma, voidaan sen stabiiliutta tarkastella vektorin $\mathbf{x}$ 2-normin $||\mathbf{x}||_\mathbf{2}= (\mathbf{x^T x})^{\mathbf{1/2}}$ ja matriisinormin $||\mathbf{A}||_\mathbf{2}=$ max$_{\mathbf{x}\neq\mathbf{0}}||\mathbf{Ax}||_\mathbf{2}/||\mathbf{x}||_\mathbf{2}$ avulla. Positiivisesti definiitille matriisille on voimassa $||\mathbf{A}||_\mathbf{2}=$ max$(\lambda_\mathbf{i})$, missä λ_i on matriisin $\mathbf{A}$ ominaisarvo. Jos Cholesky-hajotelma saatin muodostettua algoritmin avulla, muodostetuun $\hat{\mathbf{R}}$ matriisiin sisältyy virhettä liukuluvuilla laskettaessa. Muodostettu matriisi $\hat{\mathbf{R}}$ toteuttaa ehdot [4]

$$\hat{\mathbf{R}}^\mathbf{T}\hat{\mathbf{R}} = \mathbf{A} + \Delta\mathbf{A}_1 \tag{3.16}$$

$$||\Delta\mathbf{A}_1||_2 \leq c_1 n^2 u ||\mathbf{A}||_2. \tag{3.17}$$

Ylemmässä yhtälössä (3.16) matriisi $\Delta\mathbf{A}_1$ on laskennassa muodostunut virhe. Alemman yhtälön (3.17) c_1 on jokin rivimäärästä n riippuva suhteellisen pieni vakio ja u tarkoittaa tietokoneen laskenta tarkkuutta. Useimmat nykyajan tietokoneet pystyvät laskemaan tarkkuudella $2^{-53} \approx 1.1 \times 10^{-16}$. Virhetulosta (3.16) voidaan tulkita siten, että laskettuun $\hat{\mathbf{R}}$ matriisiin syntyy virhettä $\Delta\mathbf{A}_1$ verran. Tälle syntynyneelle virheelle saadaan yläraja yhtälöstä (3.17), mikä yleisesti ottaen on hyvin pieni [4].

Cholesky-hajotelmalla ratkaistu matriisiyhtälön $\mathbf{Ax} = \mathbf{b}$ ratkaisu $\hat{\mathbf{x}}$ toteuttaa ehdot

$$(\mathbf{A} + \Delta\mathbf{A}_2)\hat{\mathbf{x}} = \mathbf{b} \tag{3.18}$$

$$||\Delta\mathbf{A}_2||_2 \leq c_2 n^2 u ||\mathbf{A}||_2. \tag{3.19}$$

Saatua virhetulosta voidaan tulkita siten, että oikeasti algoritmi ratkaisee matriisin $(\mathbf{A} + \Delta\mathbf{A}_2)x = b$ ratkaisun. Matriisin $\Delta\mathbf{A}_2$ suurin mahdollinen vaikutus yhtälöön (3.18) saadaan yhtälöstä (3.19). Cholesky-hajotelman tapauksessa tätä vaikutusta voidaan pitää mitättömänä, jolloin laskettua ratkaisua $\hat{\mathbf{x}}$ voidaan pitää alkuperäisen yhtälön $\mathbf{Ax} = \mathbf{b}$ oikeana, luotettavana ratkaisuna. [4]

Cholesky-hajotelman numeerinen stabiilius seuraa epäyhtälöstä

$$r_{ij}^2 \leq \sum_{k=1}^{i} r_{kj}^2 = a_{jj}.$$

Tämä käytännössä tarkoittaa, että matriisin $\mathbf{R}$ alkiot rajautuvat alkuperäisen hajoitetun matriisin $\mathbf{A}$ diagonaalialkioiden mukaan. Eli siis esimerkiksi $\mathbf{R}$ matriisin j :nnen sarakkeen jokaisen alkion neliö on pienempää tai yhtäsuurta kuin matriisin $\mathbf{A}$ diagonaalialkio a_{jj}. Koska matriisin $\mathbf{R}$ kaikki alkiot r_{ij} ovat tällä tavalla rajoitettuja, on Cholesky-hajolma stabiili [2, s. 165].

4 CHOLESKY-HAJOTELMAN SUORITUSKYKY LU-HAJOTELMAAN VERRATTUNA

Aikaisemmin mainittiin, että jos Cholesky-hajotelma on mahdollista muodostaa jollekin sen ehdot täyttävälle matriisille, myös LU-hajotelma on muodostettavissa. Tilanne ei kuitenkaan ole sama päinvastoin eli läheskään kaikille LU-hajotelman omaaville matriiseille ei voida muodostaa Cholesky-hajotelmaa. Usein ajatellaankin, että Cholesky-hajotelma on LU-hajotelman erityistapaus, jossa symmetrisyyttä käytetään hyväksi hajotelmaa muodostettaessa. Tarkoituksena tässä luvussa ei ole käydä LU-hajotelman algoritmeja tarkasti läpi, vaan vain kertoa miten ne olennaisesti eroavat Cholesky-hajotelman vastaavista sekä miten nämä erot vaikuttavat hajotelmien suorituskykyyn.

Molemmat hajotelmat pyrkivät samaan lopputulokseen, jossa pyritään löytämään ratkaisu $\mathbf{x}$ johonkin lineaariseen yhtälöryhmään $\mathbf{Ax} = \mathbf{b}$. Ratkaisutapa molemmilla tavoilla on identtinen. Ensin hajoitetaan matriisi $\mathbf{A}$ ylä- ja alakolmio muotoihin, minkä jälkeen ratkaistaan kaksi eri matriisiyhtälöä. Choleskyn tapauksessa ratkaistaan ensin yhtälöstä $\mathbf{R^T y} = \mathbf{b}$ vektori $\mathbf{y}$, minkä jälkeen voidaan ratkaista yhtälöstä $\mathbf{Rx} = \mathbf{y}$ alkuperäisen yhtälön $\mathbf{Ax} = \mathbf{b}$ ratkaisu $\mathbf{x}$. LU-hajotelman prosessi on muutoin sama, mutta yhtälöt esitetään $\mathbf{L}$ ja $\mathbf{U}$ matriisien avulla muodossa $\mathbf{Ly} = \mathbf{b}$ ja $\mathbf{Ux} = \mathbf{y}$. Oleellinen asia koko operaation kannalta on, että olemmat hajotelmat vaativat yhtä paljon laskentaoperaatioita hajotelman jälkeisessä sijoitusvaiheessa. Tämän vuoksi laskenta-aikojen kannalta eroa eri hajotelmilla ratkaistaessa syntyy vain itse hajotelman muodostamisessa. Sijoitusvaiheen operaatioiden määrä on luokkaa n^2 [2], jolloin suurilla syötteen ko'oilla koko prosessin vaatima aika määräytyy käytetyn hajotelman ja sen laskentaoperaatioiden mukaan.

Edellisessä luvussa Cholesky-hajotelman muodostamiselle saatiin tulokseksi $n^3/3$ laskentaoperaatioita. LU-hajotelman muodostamiseen vaaditaan sen sijaan $2n^3/3$ laskentaoperaatioita [2]. LU-hajotelma muodostetaan kertomalla hajotettavaa matriisia Gaussinmatriiseilla, jotka muuttavat matriisin yläkolmiomuotoon. Tämä vaihe LU-hajotelmassa vaatii saman verran laskentaoperaatioita, kuin Cholesky-hajotelman algoritmi. Cholesky-hajotelmassa alakolmiomatriisi saatiin suoraviivaisesti transponoimalla yläkolmiomatriisi, kun taas LU-hajotelmassa tulee vielä määrittää erikseen etsittävä alakolmiomatriisi. Tämä määritys vie jälleen saman määrän operaatioita kuin yläkolmion määritys, minkä vuoksi Cholesky-hajotelma selviää puolet vähemmällä työllä. Näin ollen Cholesky-hajotelma on puolet nopeampi verrattuna LU-hajotelmaan. Molempien algoritmien vaatimat laskentaoperaatiot ovat samassa kertaluokassa n^3. Tämä tarkoittaa, että molemmat algoritmit hidastuvat yhtä nopeasti syötten koon kasvaessa. Eli kummassakin tapauksessa syötteen koon kaksinkertaistuessa algoritmien vaatimat ajat kahdeksankertaistuvat. Tämä ei kuitenkaan tarkoita, etteikö Cholesky-hajotelmaa kannattaisi käyttää, vaan nimenomaan suuria matriiseja käsiteltäessä Cholesky-hajotelmalla selviää pienemmällä työllä, vaikkakin se hidastuu suhteessa yhtänopeasti kuin LU-hajotelman algoritmi.

Toinen Cholesky-hajotelman hyvä puoli on, että se kuluttaa vähemmän tietokoneen muistia. Cholesky-hajotelman ylä- ja alakolmiomatriisin tallennukseen kuluu muistia puolet vähemmän verrattuna LU-hajotelmaan. LU-hajotelmassa sekä ylä- että alakolmiomatriisi tulee tallentaa muistiin, kun taas Choleskyn tapauksessa riittää pelkän yläkolmion tallennus sen transpoosiominaisuuden vuoksi.

5 YHTEENVETO

Työssä tutkittiin Cholesky-hajotelmaa ja todistettiin sen olemassaolo matriiseille, jotka ovat reaalisia, symmetrisiä ja positiivisesti definiittejä. Tämän lisäksi työssä esiteltiin yksi tapa määrittää Cholesky-hajotelma matriisille, sekä tarkasteltiin kyseisen tavan algoritmia ja sen tehokkuutta. Algoritmin tehokkuutta ja suorituskykyä vertailtiin LU-hajotelmaan sekä perusteltiin, miksi Cholesky-hajotelma on näistä kahdesta kaikilla tavoin optimaalimpi, jos sitä vain pystyy soveltamaan. Työssä tarkasteltu algoritmi on hyvin yleisesti käytetty tapa muodostaa Cholesky-hajotelma. Lisäksi on olemassa tapoja, joissa hajotelman muodostamiseen käytetään matriisiblokkeja ja rekursiota. Näihin tapoihin ei tässä työssä perehdytty, mutta mainitsemisen arvoista kuitenkin on, että Cholesky-hajotelma on joissakin tapauksissa mahdollista muodostaa tehokkaammin tällaisilla blokkimatriiseja hyödyntävillä algoritmeilla.

Kaiken kaikkiaan Cholesky-hajotelma on hyödyllinen ja tehokas työkalu lineaaristen yhtälöryhmien ratkaisuun, jos yhtälöryhmän kerroinmatriisi kattaa Cholesky-hajotelman välttämättömät ehdot. Vaatimukset reaalisuus, symmetrisyys ja positiivisesti definiittisyys ovat kerroinmatriisille kuitenkin melko kovat, sillä satunnainen matriisi ei toteuta näitä vaatimuksia juuri koskaan. Näin ollen Cholesky-hajotelman käytön voidaan ajatella kohdistuvan tiettyihin, erikoistapauksina pidettäviin, tilanteisiin. Toisaalta matriisin reaalisuus ei ole välttämätöntä, vaikka työssä matriisit rajattiin reaalisiksi. Kompleksisille matriiseille voidaan myös muodostaa Cholesky-hajotelma, mutta tällaisten tapauksien tarkastelua ei tähän työhön sisällytetty.

LÄHDELUETTELO

[1] T. S. Blyth ja E. F. Robertson. *Basic Linear Algebra*. Springer, 2002, s.61.

[2] G. H. Golub ja C. F. Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012, s.153–176.

[3] N. J. Higham. *Accuracy and stability of numerical algorithms*. Vol. 80. Siam, 2002, s.195–211.

[4] N. J. Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics* 1.2 (2009).

[5] P. D. Powell. Calculating determinants of block matrices. *arXiv preprint arXiv:1112.4379* (2011).

[6] D. S. Watkins. *Fundamentals of matrix computations*. John Wiley & Sons, 2010, s.33–94.