TEKNILLINEN KORKEAKOULU
Informaatio- ja luonnontieteiden tiedekunta

Allan Perämäki

Reaalilineaaristen yhtälöryhmien iteratiiviset menetelmät

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin tutkintoa varten teknillisen fysiikan koulutusohjelmassa.

Espoossa, 8.12.2009

Työn valvoja: professori Timo Eirola
Työn ohjaaja: akatemiatutkija Marko Huhtanen

Tekijä:	Allan Perämäki
Koulutusohjelma:	Teknillinen fysiikka
Pääaine:	Matematiikka
Sivuaine:	Tietoliikenneohjelmistot

Työn nimi:	Reaalilineaaristen yhtälöryhmien iteratiiviset menetelmät
Title in English:	Iterative Methods for Real-Linear Systems of Equations
Professuurin koodi ja nimi:	Mat-1 Matematiikka
Työn valvoja:	Professori Timo Eirola
Työn ohjaaja:	Akatemiatutkija Marko Huhtanen

Tiivistelmä:

Työssä esitellään ratkaisumenetelmiä yhtälöille muotoa $\boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, missä $\boldsymbol{M}$ ja $\boldsymbol{M}_{\#}$ ovat kompleksisia $n \times n$-matriiseja, $\boldsymbol{b}$ annettu vektori ja $\boldsymbol{z}$ ratkaistava vektori. Tällainen yhtälö voitaisiin ratkaista muuttamalla se ensiksi reaaliseksi $2n \times 2n$-kokoiseksi lineaariseksi yhtälöksi $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$, mutta tässä työssä menetelmät perustuvat alkuperäiseen muotoon.

Reaalilineaariset operaattorit ovat kuvauksia $\boldsymbol{z} \mapsto \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}}$. Tässä työssä nämä esitetään taulukkomuodossa matriisien tapaan ja niille annetaan lähteen [1] mukainen reaalilineaarinen LU-hajotelma, Householderin muunnos ja QR-hajotelma. Householderin muunnoksen laskenta-algoritmi annetaan numeerista stabiiliutta parantavalla muutoksella. Lähteessä [1] mainitun LU-hajotelman tukialkion singulaarisuuden ongelman ratkaisuksi ehdotetaan uutta menetelmää.

Matriisien Krylovin aliavaruudet ja näihin perustuvia lineaaristen yhtälöryhmien iteratiivisia ratkaisumenetelmiä esitellään lähdettä [8] noudatellen. Lähteen [1] mukainen reaalilineaarinen GMRES-menetelmä yhtälölle $\kappa\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, missä κ on kompleksiluku, käydään läpi ja lisäksi tapauksille $\boldsymbol{M}_{\#}^{T} = (-)\boldsymbol{M}_{\#}$ tuodaan esiin Lanczos-tyyppiseen iterointiin perustuvat menetelmät. Ensiksi mainitussa menetelmässä tarvittavan Householderin muunnoksen tilalle tarjotaan uutena mahdollisuutena käyttää reaalilineaarisia Givens-rotaatioita. Yleisille reaalilineaarisille yhtälöille ei saada GMRESia, mutta joitakin Galerkinin menetelmään perustuvia iteratiivisia menetelmiä kokeillaan mukaanlukien lähteessä [1] mainittu. Matriisien epätäydelliset LU-hajotelmat (ILU) pohjustusmenetelminä yleistetään reaalilineaarisille operaattoreille. Näistä esimerkkeinä annetaan ILU(0)- ja ILUT-menetelmät.

Numeerisia kokeita varten diskretoidaan sähköisessä impedanssitomografiassa esiintyvää $\overline{\partial}$-yhtälöä vastaava integraaliyhtälö päätyen yhtälöön $\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{1}$ lähdettä [5] noudattaen. MATLABilla suoritetut laskennat osoittavat reaalilineaariset ratkaisumenetelmät tehokkaammiksi kuin GMRES vastaavalle reaaliselle $2n \times 2n$-yhtälöryhmälle. Aikaisemmin ei ole suoraan mainittu edellä olevan yhtälö muuttamista $\mathbb{C}$-lineaariseen muotoon $(\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}})\boldsymbol{z} = \boldsymbol{1} - \boldsymbol{M}_{\#}\boldsymbol{1}$. Matriisin $\boldsymbol{M}_{\#}$ ollessa peräisin edellä mainitun integraaliyhtälön diskretoinnista, on numeeristen kokeiden perusteella ratkaisu tästä muodosta suositeltavaa.

Lukijalta edellytetään lineaarialgebran alkeiden hallintaa. Liite A sisältää matriisien osalta käytetyt määritelmät.

Sivumäärä: 84	Avainsanat:	reaalilineaarinen, Krylov, iteraatio, Galerkin, GMRES, pohjustus, ILU, impedanssitomografia

Täytetään tiedekunnassa

Hyväksytty: Kirjasto:

Author:	Allan Perämäki
Degree Programme:	Engineering Physics
Major subject:	Mathematics
Minor subject:	Data Communications Software

Title:	Iterative Methods for Real-Linear Systems of Equations
Title in Finnish:	Reaalilineaaristen yhtälöryhmien iteratiiviset menetelmät
Chair:	Mat-1 Mathematics
Supervisor:	Professor Timo Eirola
Instructor:	Academy Research Fellow Marko Huhtanen

Abstract:

This thesis presents solution methods for equations of the form $\boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, where $\boldsymbol{M}$ and $\boldsymbol{M}_{\#}$ are complex $n \times n$-matrices, $\boldsymbol{b}$ a given vector and $\boldsymbol{z}$ the vector to be solved. Such an equation could be solved by first rewriting it as a real $2n \times 2n$ system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$, but the methods in this thesis are based on the original formulation.

Real-linear operators are mappings $\boldsymbol{z} \mapsto \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}}$. These can be represented in a table form not unlike matrices and real-linear LU-decomposition, Householder transformation and QR-decomposition are defined following [1]. The Householder transformation is modified to improve numerical stability. A new strategy to avoid the breakdown mentioned in [1] in the real-linear LU-decomposition is suggested.

The Krylov subspaces of matrices and some solutions methods based on these are introduced roughly following [8]. The real-linear GMRES method given in [1] for the equation $\kappa\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, where κ is a complex number, is reconsidered and additionally methods for the cases $\boldsymbol{M}_{\#}^{T} = (-)\boldsymbol{M}_{\#}$ based on Lanczos-type iteration is presented. New real-linear Givens rotations provide an alternative to using Householder transformations during computations of the real-linear GMRES. There's no GMRES for real-linear equations of the general form, but some iterative Galerkin methods are tried including the one mentioned in [1]. The incomplete LU-decompositions (ILU) for matrices as preconditioning methods are generalized to real-linear operators. ILU(0) and ILUT methods are given as examples.

Following [5], the integral equation corresponding to the $\overline{\partial}$-equation arising in electrical impedance tomography is discretized resulting in the equation $\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{1}$. Computations with `MATLAB` provide evidence of the effectiveness for real-linear methods in comparison to the corresponding real $2n \times 2n$ system with GMRES. Previously, there's no direct mention of transforming to the $\mathbb{C}$-linearized equation $(\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}})\boldsymbol{z} = \boldsymbol{1} - \boldsymbol{M}_{\#}\boldsymbol{1}$. When the matrix $\boldsymbol{M}_{\#}$ arises from the discretization of the mentioned integral equation numerical experiments suggest that using this formulation is recommended.

The thesis is self-contained requiring only the basics of linear algebra. Appendix A contains the required definitions concerning matrices.

Number of pages: 84	Keywords:	real linear, Krylov, iteration, Galerkin, GMRES, preconditioning, ILU, impedance tomography

Faculty fills

Approved:	Library code:

Sisällys

1 Johdanto **1**

1.1 Reaalilineaariset operaattorit 3

1.1.1 Reaalimatriisiesitys 4

1.1.2 Yhdistetty kuvaus ja käänteisoperaattori 6

1.1.3 Liitto-operaattori 7

1.2 QR-hajotelma 8

2 Suora ratkaiseminen ja LU-hajotelma **12**

2.1 $\mathbb{R}$-lineaaristen operaattoreiden LU-hajotelma 12

2.2 Osittaistuenta 14

2.3 Tuenta $\mathbb{R}$-lineaariselle LU-hajotelmalle 16

2.3.1 $\mathbb{R}$-lineaarisesti tuetun LU-hajotelman laskeminen 19

3 Iteratiivinen ratkaiseminen **20**

3.1 Johdanto 20

3.2 Petrov-Galerkinin menetelmä 21

3.3 Matriisien Krylovin aliavaruudet 23

3.4 GMRES-menetelmä 25

3.4.1 Uudelleenkäynnistys 28

3.5 $\mathbb{R}$-lineaariset menetelmät 28

3.6 Krylovin aliavaruudet 29

3.7 $\mathbb{R}$-lineaarinen täyden ortogonalisoinnin menetelmä 33

3.7.1 Menetelmä symmetriselle $\boldsymbol{M}_{\#}$ 33

3.7.2 Menetelmä vinosymmetriselle $\boldsymbol{M}_{\#}$ 36

3.8 $\mathbb{R}$-lineaarinen GMRES operaattorille $\boldsymbol{\mathcal{M}}_{\kappa}$ 38

3.8.1 MINRES symmetriselle $\boldsymbol{M}_{\#}$ 42

3.9 $\mathbb{C}$-linearisoitu yhtälö 42
3.10 Yleisen $\mathbb{R}$-lineaarisen operaattorin menetelmät 44

4 ILU-pohjustus **47**
4.1 Johdanto 47
4.2 Matriisiyhtälöiden ILU-pohjustus 48
4.2.1 ILU-hajotelma ennaltavalitulla nollakuviolla 48
4.2.2 ILU-hajotelma mukautetulla nollakuviolla 49
4.3 $\mathbb{R}$-lineaaristen yhtälöiden ILU-pohjustus 50
4.3.1 $\mathbb{R}$-lineaarinen ILU-hajotelma ennaltavalitulla nollakuviolla . . 52
4.3.2 $\mathbb{R}$-lineaarinen ILU-hajotelma mukautetulla nollakuviolla . . . 53

5 Numeerisia kokeita **56**
5.1 GMRES kokeita 56
5.2 GMRES impedanssitomografiassa 57
5.2.1 $\overline{\partial}$-yhtälön ratkaiseminen 59
5.2.2 Brown-Uhlmann menetelmä 64
5.2.3 Kokeet 65
5.2.4 $\overline{\partial}$-yhtälö satunnaisella kerroinfunktiolla 65
5.3 Aaltoyhtälö 68
5.4 Epälineaarinen Schrödingerin yhtälö 69

Yhteenveto **73**

Kirjallisuutta **74**

A Matriisit **76**

B MATLAB-ohjelmalistaukset **78**
B.1 Kohta 2.3.1 78
B.2 Kohta 4.2.1 80
B.3 Kohta 4.2.2 81
B.4 Kohta 4.3.1 82
B.5 Algoritmi 4.3.2 83

Luku 1

Johdanto

Matriisien teorian alkuperänä on lineaaristen yhtälöryhmien ratkaisemisessa käytetyt taulukkomuodot. Gaussin eliminaationa tunnettu menetelmä esiintyy jo kolmannelta tai toiselta vuosisadalta eKr peräisin olevassa kiinalaisessa kokoelmassa *Tšiu-tšang suan-su* (Yhdeksän matematiikan taitoja käsittelevää lukua). Japanissa 1700-luvun lopulla Seki Kowa päätyi determinantin käsitteeseen näiden taulukkomuotojen alkeisoperaatioiden pohjalta. Näistä riippumatta samoihin aikoihin Saksassa Gottfried W. Leibniz kehitti myös determinantin, joskin nähtävästi hän käsitteli vain kahden ja kolmen yhtälön ja muuttujan yhtälöryhmiä.

Leibnizin determinantit ja ratkaisutapa kuitenkin vaipuivat unholaan. Hänestä riippumatta sveitsiläinen Gabriel Cramer keksi 1750 hänen nimeään nykyäänkin kantavan yleisen säännön ratkaista lineaarisia yhtälöryhmiä determinanttien avulla. Sääntö teki determinanteista suositun ratkaisutavan vaikka luultavasti se olikin kehitetty jo tunnetun eliminaatiomenetelmän pohjalta. Tuolloin ei kuitenkaan yleensä edes yritetty ratkaista kovin suuria yhtälöryhmiä.

Carl F. Gaussin taivaanmekaniikkaa käsittelevät julkaisut 1800-luvun alkupuolella sisälsivät pienimmän neliösumman tehtävän ratkaisumenetelmän. Siinä esiintyvän normaaliyhtälön kertoimet muodostavat nykykielellä ilmaisten positiividefiniitin symmetrisen matriisin. Gauss ratkaisi yhtälön eliminointimenetelmällä, joka hyödynsi näitä ominaisuuksia laskutyön nopeuttamiseksi. Esimerkkinä Gauss laskee korjauksen Aurinkokunnan toiseksi suurimman asteroidin Pallaksen rataelementeille, missä ratkaistava normaaliyhtälö sisältää kuusi yhtälöä ja kuusi muuttujaa. Gaussin eliminoinnin nimitys juontaa juurensa näistä julkaisuista.

Matriisi-nimityksen otti käyttöön James J. Sylvester 1850. Hän käytti näitä taulukoita determinanttien teoriassa. Varsinaisen matriisien määritelmän teki Arthur Cayley joitakin vuosia myöhemmin. Hän määritteli matriisien tulon pohtimalla tekijöitä vastaavien lineaarikuvausten yhdistettä. Cayleyn töissä esiintyy kaikki matriisialgebran perusominaisuudet kuten skalaarilla kertoiminen, matriisien yhteenlasku, epäkommutoiva tulo, singulaarisuus ja yhteys aikaisempaan determinanttien teoriaan.

1940-luvulle mentäessä olivat erilaiset matriisien hajotelmat (LU, Cholesky) käytössä. Mekaanisten pöytälaskukoneiden avulla oli mahdollista ratkaista 10-20 yhtälön ja muuttujan lineaarisia yhtälöryhmiä Gaussin eliminoinnilla. Rajallisen laskentatarkkuuden ja sen pyöristysvirheiden vaikutusta eliminoinnissa ei kuitenkaan tunnettu ja

monet matemaatikot olivatkin pessimistisiä Gaussin eliminoinnin ja suurten yhtälöryhmien suhteen. Aiemmasta pessimistisyydestään huolimatta John von Neumann ja Herman Goldstine sekä Alan Turing julkaisivat 40-luvun lopulla eliminointiprosessin virhearvioita, jotka hälvensivät epäilyksiä pyöristysvirheiden katastrofaalisuudesta. James H. Wilkinson julkaisi 1961 paremmat virhearviot ja otti käyttöön nimitykset osittaistuenta ja täystuenta (näitä menetelmiä oli käytetty jo aiemmin, mm. Wilkinson itse ratkaistessaan pöytälaskukoneella 18 yhtälön ryhmää 1946).

Tietokoneiden kehityksen myötä on Gaussin eliminoinnilla tuhansien yhtälöiden tehtävät nykyään helppo ratkaista. Supertietokoneilla suurimmat tiheiden kerroinmatriisien yhtälöryhmät ovat sisältäneet yli miljoona yhtälöä. Hyvin monissa käytännön ongelmissa kerroinmatriisit kuitenkin koostuvat suurelta osin nollista, minkä johdosta tämän hyödyntämisen mahdollistavat iteratiiviset menetelmät ovat nykyään suosittuja. Iteratiivisia menetelmiä lineaarisen yhtälöryhmän ratkaisemiseksi on esitetty jo 1800-luvulla. Näitä ovat mm. Gauss-Seidelin iteraatio ja Jacobin iteraatio. Tällöin kyseessä ei kuitenkaan ollut matriisin harvuusrakenteen hyödyntäminen, vaan tarkan laskutyön helpottaminen. 1950-luvulla tietokoneilla iteroitiin mm. ylirelaksaatiomenetelmällä (SOR), jolla vuonna 1960 kyettiin rutiininomaisesti ratkaisemaan kaksiulotteisia Laplace-tyyppisiä 20000 yhtälön ryhmiä. Yleisten yhtälöryhmien ratkaiseminen Gaussin eliminoinnilla oli tuolloin mahdollista noin 100:lle yhtälölle.

Liittogradienttimenetelmä ja Lanczosin menetelmä syntyivät 1950-luvun alussa ratkaisemaan reaalisen symmetrisen matriisin yhtälöryhmiä. Tarkoilla laskutoimituksilla näillä menetelmillä on ominaisuus, että ratkaisu saavutetaan yhtälöiden lukumäärä vastaavalla iteraatiokierroksella. Tästä syystä näitä pidettiin aluksi mahdollisina suorina menetelminä. Numeeriset kokeet hankalilla yhtälöryhmillä osoittivat ettei edellä mainittu tarkan aritmetiikan ominaisuus pätenyt tietokonearitmetiikassa. Liittogradienttimenetelmä saavutti suuren suosion vasta 70-luvulla, kun sitä alettiin pitämään aitona iteratiivisena menetelmänä ja se yhdistettiin soveltuvaksi käyttöön harvoille matriiseille. Lisäksi tuolloin tiedettiin, että iteraation suppenemisnopeus riippuu matriisin ominaisarvojakaumasta. Edellä mainitut menetelmät voidaan nähdä Krylovin (venäläisen Aleksei Krylovin vuonna 1931 julkaiseman menetelmän perusteella) aliavaruusmenetelminä, joiden soveltaminen perustuu matriisi-vektoritulojen laskemiseen. Näistä mainittakoon vielä myös epäsymmetrisille matriiseille soveltuva vuonna 1986 Yousef Saadin ja Martin Schultzin esittämä GMRES.

Krylovin aliavaruusmenetelmien kanssa ryhdyttiin 70-luvulla yleisesti käyttämään myös pohjustusmenetelmiä suppenemisen nopeuttamiseksi. Yhtälöryhmän kerroinmatriisin ajatellaan tällöin olevan kerrottu sen likimääräisellä käänteismatriisilla, jonka laskemisen on oltava nopeaa. Näistä suosittuja ovat Gaussin eliminaatioon perustuva ILU (incomplete LU) ja sen muunnelmat.

Osittaisdifferentiaaliyhtälöiden ja integraaliyhtälöiden diskretoinnista saadaan usein kerroinmatriiseja, joilla on harvuusrakenteensa tai muun rakenteen johdosta nopea laskea matriisi-vektori tuloja. Täten Krylovin aliavaruusmenetelmät soveltuvat niiden ratkaisuun. Kompleksisten yhtälöiden diskretoinnista saadaan kompleksiluvuista koostuvia kerroinmatriiseja ja ratkaisuja eikä alun perin reaalisille matriiseille kehitettyjen menetelmien suora soveltaminen ole aina järkevää. Liittogradienttimenetelmä voidaan yleistää näille, kunhan matriisi on hermiittinen. Samoin GMRESin voi yleistää suoraviivaisesti kompleksimatriiseille. Kuitenkin on olemassa tapauksia (kompleksinen Helmholtzin yhtälö), joissa kerroinmatriisi on symmetrinen ja

kompleksinen. Tällöin voitaisiin käyttää GMRESia, mutta näille on kehitetty myös symmetristä rakennetta hyödyntävä liittogradienttimenetelmän-tyyppinen menetelmä.

Tämän diplomityön aiheena on matriisien sijaan reaalilineaariset operaattorit ja näiden muodostamat yhtälöryhmät. Tyyppiesimerkkinä on sähköisessä impedanssitomografiassa esiintyvän (kompleksisen) integraaliyhtälön diskretoinnista saatu reaalilineaarinen yhtälö. Seuraavaksi tutustutaan operaattoreihin tarkemmin.

Edellä oleva tiivistelmä lineaaristen yhtälöryhmien ratkaisemisen historiasta on laadittu lähteiden [12], [13], [14] ja [15] pohjalta.

1.1 Reaalilineaariset operaattorit

Lineaarialgebrassa tutkitaan äärellisdimensioisia vektoriavaruuksia ja niiden välisiä lineaarisia kuvauksia. Lineaarioperaattorilla A kompleksilukukertoimisten vektoriavaruuksien V ja W välillä, $A : V \to W$, on ominaisuudet $A(v_1 + v_2) = Av_1 + Av_2$ ja $A(\alpha v_1) = \alpha A v_1$ kaikilla vektoreilla $v_1, v_2 \in V$ ja $\alpha \in \mathbb{C}$. Kun avaruuksiin V ja W kiinnitetään kanta, voidaan operaattorille A muodostaa sitä esittävä matriisi.

Tässä työssä käsitellään operaattoreita $\boldsymbol{\mathcal{M}} : V \to W$, joilla on ominaisuudet

$$\begin{aligned} \boldsymbol{\mathcal{M}}(v_1 + v_2) &= \boldsymbol{\mathcal{M}}(v_1) + \boldsymbol{\mathcal{M}}(v_2), && \text{kaikilla vektoreilla } v_1, v_2 \in V \\ \boldsymbol{\mathcal{M}}(\alpha v) &= \alpha \boldsymbol{\mathcal{M}}(v), && \text{kaikilla vektoreilla } v \in V \text{ ja luvuilla } \alpha \in \mathbb{R}. \end{aligned}$$

Jälkimmäinen ominaisuus vaaditaan nyt vain reaaliluvuille α, mutta avaruudet V ja W ovat edelleen kompleksikertoimisia. Kun avaruuksiin kiinnitetään kannat, voidaan tällainen operaattori esittää kahdella matriisilla. Olkoon $v_1, v_2, \ldots, v_n$ avaruuden V kanta ja $w_1, w_2, \ldots, w_m$ avaruuden W kanta. Määritellään $m \times n$-matriisit $A = (a_{jk})$ ja $B = (b_{jk})$ siten, että

$$\sum_{j=1}^{m} a_{jk} w_j = \tfrac{1}{2}\big(\boldsymbol{\mathcal{M}}(v_k) - i\boldsymbol{\mathcal{M}}(iv_k)\big), \qquad \sum_{j=1}^{m} b_{jk} w_j = \tfrac{1}{2}\big(\boldsymbol{\mathcal{M}}(v_k) + i\boldsymbol{\mathcal{M}}(iv_k)\big).$$

Olkoon sitten $v \in V$ ja $v = \sum_{k=1}^{n} c_k v_k$. Tällöin

$$\begin{aligned} \boldsymbol{\mathcal{M}}(v) &= \sum_{k=1}^{n} \boldsymbol{\mathcal{M}}(c_k v_k) = \sum_{k=1}^{n} \boldsymbol{\mathcal{M}}\big(\tfrac{1}{2}(c_k + \overline{c}_k)v_k + \tfrac{1}{2i}(c_k - \overline{c}_k) i v_k\big) \\ &= \sum_{k=1}^{n} \big(\tfrac{1}{2}(c_k + \overline{c}_k)\boldsymbol{\mathcal{M}}(v_k) + \tfrac{1}{2i}(c_k - \overline{c}_k)\boldsymbol{\mathcal{M}}(iv_k)\big) \\ &= \sum_{k=1}^{n} \big(\tfrac{1}{2}(\boldsymbol{\mathcal{M}}(v_k) - i\boldsymbol{\mathcal{M}}(iv_k))c_k + \tfrac{1}{2}(\boldsymbol{\mathcal{M}}(v_k) + i\boldsymbol{\mathcal{M}}(iv_k))\overline{c}_k\big) \\ &= \sum_{k=1}^{n}\sum_{j=1}^{m} a_{jk} c_k w_j + \sum_{k=1}^{n}\sum_{j=1}^{m} b_{jk}\overline{c}_k w_j = \sum_{j=1}^{m}\Big(\sum_{k=1}^{n} a_{jk} c_k + \sum_{k=1}^{n} b_{jk}\overline{c}_k\Big) w_j. \end{aligned}$$

Näin ollen $\boldsymbol{\mathcal{M}}(v)$ saadaan laskemalla $Ac + B\overline{c}$, missä c on v:n komponenttien muodostoma pystyvektori. Tämä työ keskittyy numeeriseen laskentaan tällaisilla operaattoreilla, joten abstrakteja vektoriavaruuksia ei jäljempänä enää käsitellä. Asetetaankin seuraava määritelmä.

Määritelmä 1.1.1. *Olkoot* $\boldsymbol{M}$ *ja* $\boldsymbol{M}_{\#}$ *kompleksisia* $m \times n$*-matriiseja. Kuvausta*

$$\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^m, \qquad \boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\bar{\boldsymbol{z}}$$

kutsutaan $\mathbb{R}$*-lineaariseksi (reaalilineaariseksi) operaattoriksi. Matriisia* $\boldsymbol{M}$ *kutsutaan* $\boldsymbol{\mathcal{M}}$*:n lineaariseksi osaksi ja matriisia* $\boldsymbol{M}_{\#}$ $\boldsymbol{\mathcal{M}}$*:n antilineaariseksi osaksi.*

$\mathbb{R}$-lineaarista operaattoria $\boldsymbol{\mu} : \mathbb{C} \to \mathbb{C}$ kutsutaan $\mathbb{R}$-lineaariseksi skalaarioperaattoriksi. Kun $\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\bar{\boldsymbol{z}}$ on $\mathbb{R}$-lineaarinen, niin voidaan määritellä skalaarioperaattorit $\boldsymbol{\mu}_{ij} : \mathbb{C} \to \mathbb{C}$, $\boldsymbol{\mu}_{ij}(z) = (\boldsymbol{M})_{ij} z + (\boldsymbol{M}_{\#})_{ij}\bar{z}$. Operaattori $\boldsymbol{\mathcal{M}}$ voidaan silloin ilmaista $m \times n$-muodossa

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \boldsymbol{\mu}_{11} & \boldsymbol{\mu}_{12} & \boldsymbol{\mu}_{13} & \cdots & \boldsymbol{\mu}_{1n} \\ \boldsymbol{\mu}_{21} & \boldsymbol{\mu}_{22} & \boldsymbol{\mu}_{23} & \cdots & \boldsymbol{\mu}_{2n} \\ \boldsymbol{\mu}_{31} & \boldsymbol{\mu}_{32} & \boldsymbol{\mu}_{33} & \cdots & \boldsymbol{\mu}_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mu}_{m1} & \boldsymbol{\mu}_{m2} & \boldsymbol{\mu}_{m3} & \cdots & \boldsymbol{\mu}_{mn} \end{bmatrix}, \tag{1.1}$$

$$\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{\mu}_{11}(z_1) + \boldsymbol{\mu}_{12}(z_2) + \cdots + \boldsymbol{\mu}_{1n}(z_n) \\ \boldsymbol{\mu}_{21}(z_1) + \boldsymbol{\mu}_{22}(z_2) + \cdots + \boldsymbol{\mu}_{2n}(z_n) \\ \boldsymbol{\mu}_{31}(z_1) + \boldsymbol{\mu}_{32}(z_2) + \cdots + \boldsymbol{\mu}_{3n}(z_n) \\ \vdots \\ \boldsymbol{\mu}_{m1}(z_1) + \boldsymbol{\mu}_{m2}(z_2) + \cdots + \boldsymbol{\mu}_{mn}(z_n) \end{bmatrix}, \tag{1.2}$$

missä $\boldsymbol{z} = (z_1, z_2, \ldots, z_n) \in \mathbb{C}^n$. Tällöin merkitään lyhyesti $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mu}_{ij})$. Näille taulukkomuodoille käytetään vapaasti matriiseista peräisin olevia tuttuja nimityksiä. Esimerkiksi yläkolmio-operaattori tarkoittaa operaattoria, jolle $\boldsymbol{\mu}_{ij} = 0$ kaikilla $i > j$. Käytetään myös nk. `MATLAB`-notaatiota. Esimerkiksi $\boldsymbol{\mathcal{M}}_{2:4,*}$ on operaattori, joka muodostuu $\boldsymbol{\mathcal{M}}$:n toisesta, kolmannesta ja neljännestä rivistä. Kun $\boldsymbol{\mathcal{M}}$ on rivi- tai sarakeoperaattori, niin esim. $\boldsymbol{\mathcal{M}}_{1:4}$ on rivi- tai sarakeoperaattori koostuen $\boldsymbol{\mathcal{M}}$:n neljästä ensimmäisestä skalaarioperaattorista ja $\boldsymbol{\mathcal{M}}_2$ on $\boldsymbol{\mathcal{M}}$:n toinen skalaarioperaattori.

1.1.1 Reaalimatriisiesitys

$\mathbb{R}$-lineaarinen operaattori $\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\bar{\boldsymbol{z}}$ voidaan samaistaa reaalisen $2m \times 2n$-matriisin kanssa kahdella luonnollisella tavalla. Numeerisessa laskennassa tätä ei välttämättä kannata tehdä, koska matriisien $\boldsymbol{M}$ ja $\boldsymbol{M}_{\#}$ rakenne menetetään. Joitakin reaalilineaaristen operaattoreiden ominaisuuksia tästä kuitenkin nähdään.

Merkitään $\boldsymbol{z} = \boldsymbol{x} + i\boldsymbol{y} \in \mathbb{C}^n$, missä $\boldsymbol{x} = \mathrm{Re}(\boldsymbol{z})$ on $\boldsymbol{z}$:n reaaliosa ja $\boldsymbol{y} = \mathrm{Im}(\boldsymbol{z})$ imaginääriosa. Samaistetaan sitten $\boldsymbol{z} \in \mathbb{C}^n$ ja vektori $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) \in \mathbb{R}^{2n}$. Laskemalla saadaan

$$\begin{aligned} \boldsymbol{\mathcal{M}}(\boldsymbol{z}) &= \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\bar{\boldsymbol{z}} \\ &= \big(\mathrm{Re}(\boldsymbol{M}) + i\,\mathrm{Im}(\boldsymbol{M})\big)(\boldsymbol{x} + i\boldsymbol{y}) + \big(\mathrm{Re}(\boldsymbol{M}_{\#}) + i\,\mathrm{Im}(\boldsymbol{M}_{\#})\big)(\boldsymbol{x} - i\boldsymbol{y}) \\ &= \big(\mathrm{Re}(\boldsymbol{M}) + \mathrm{Re}(\boldsymbol{M}_{\#})\big)\boldsymbol{x} + \big(-\mathrm{Im}(\boldsymbol{M}) + \mathrm{Im}(\boldsymbol{M}_{\#})\big)\boldsymbol{y} + \\ &\quad i\Big(\big(\mathrm{Im}(\boldsymbol{M}) + \mathrm{Im}(\boldsymbol{M}_{\#})\big)\boldsymbol{x} + \big(\mathrm{Re}(\boldsymbol{M}) - \mathrm{Re}(\boldsymbol{M}_{\#})\big)\boldsymbol{y}\Big). \end{aligned}$$

Merkitään vektoria $\mathcal{M}(z) = u + iv$, missä u ja v ovat $\mathcal{M}(z)$:n reaali- ja imaginääriosat. Kun $\mathbb{C}^m$:n vektorit samaistetaan $\mathbb{R}^{2m}$:n vektoreiden kanssa vastaavalla tavalla kuin yllä tehtiin $\mathbb{C}^n$:lle, niin $\mathcal{M}(z) = u + iv$ voidaan esittää muodossa

$$\begin{bmatrix} \mathrm{Re}(M) + \mathrm{Re}(M_\#) & -\mathrm{Im}(M) + \mathrm{Im}(M_\#) \\ \mathrm{Im}(M) + \mathrm{Im}(M_\#) & \mathrm{Re}(M) - \mathrm{Re}(M_\#) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}. \tag{1.3}$$

Tässä esiintyvä $2m \times 2n$-matriisi on haettu $\mathcal{M}$:n kanssa samaistettava reaalinen matriisi.

Toinen luonnollinen tapa samaistaa $\mathbb{C}^n$:n vektorit $\mathbb{R}^{2n}$:n vektorien kanssa on käyttää vektoria
$(x_1, y_1, x_2, y_2, \ldots, x_n, y_n) \in \mathbb{R}^{2n}$, kun $z = x + iy$. Käytetään vastaavaa järjestystä $\mathbb{C}^m$:ssä. Kun sitten muodostetaan ensimmäisen tavan mukaisesti reaalinen 2×2-matriisi m_{ij} jokaiselle (1.1):n skalaarioperaattorille μ_{ij}, niin operaattoria $\mathcal{M}$ vastaa reaalinen $2m \times 2n$-matriisi

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & \ldots & m_{1n} \\ m_{21} & m_{22} & m_{23} & \ldots & m_{2n} \\ m_{31} & m_{32} & m_{33} & \ldots & m_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & m_{m3} & \ldots & m_{mn} \end{bmatrix}. \tag{1.4}$$

Esimerkki 1.1.2. Olkoon $\xi_1 : \mathbb{C} \to \mathbb{C}^3$ 3×1-operaattori siten, että $\xi_1(z) = (z, 0, 0)$. Kun merkitään nolla- ja yksikköskalaarioperaattoreita $\mathbf{0}(z) = 0$ ja $\mathbf{1}(z) = z$, niin ξ_1:n 3×1-muoto ja kohdan (1.4) mukainen matriisi ovat

$$\xi_1 = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \text{ja} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Esimerkki 1.1.3. Olkoot $M = \begin{bmatrix} 1+2i & 3+4i \\ 5+6i & 7+8i \end{bmatrix}$ ja $M_\# = \begin{bmatrix} 9+10i & 11+12i \\ 13+14i & 15+16i \end{bmatrix}$.
Kohdan (1.1) mukaiset skalaarioperaattorit ovat

$$\mu_{11}(z) = (1+2i)z + (9+10i)\bar{z} \qquad \mu_{12}(z) = (3+4i)z + (11+12i)\bar{z}$$
$$\mu_{21}(z) = (5+6i)z + (13+14i)\bar{z} \qquad \mu_{22}(z) = (7+8i)z + (15+16i)\bar{z}.$$

Kohdan (1.3) mukainen reaalinen matriisi on

$$\begin{bmatrix} \mathrm{Re}(M) + \mathrm{Re}(M_\#) & -\mathrm{Im}(M) + \mathrm{Im}(M_\#) \\ \mathrm{Im}(M) + \mathrm{Im}(M_\#) & \mathrm{Re}(M) - \mathrm{Re}(M_\#) \end{bmatrix} = \begin{bmatrix} 10 & 14 & 8 & 8 \\ 18 & 22 & 8 & 8 \\ 12 & 16 & -8 & -8 \\ 20 & 24 & -8 & -8 \end{bmatrix}$$

ja kohdan (1.4) mukaiset matriisit ovat

$$m_{11} = \begin{bmatrix} 10 & 8 \\ 12 & -8 \end{bmatrix} \quad m_{12} = \begin{bmatrix} 14 & 8 \\ 16 & -8 \end{bmatrix} \qquad \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \begin{bmatrix} 10 & 8 & 14 & 8 \\ 12 & -8 & 16 & -8 \\ 18 & 8 & 22 & 8 \\ 20 & -8 & 24 & -8 \end{bmatrix}.$$
$$m_{21} = \begin{bmatrix} 18 & 8 \\ 20 & -8 \end{bmatrix} \quad m_{22} = \begin{bmatrix} 22 & 8 \\ 24 & -8 \end{bmatrix}$$

1.1.2 Yhdistetty kuvaus ja käänteisoperaattori

$\mathbb{R}$-lineaarisista operaattoreista $\boldsymbol{\mathcal{M}} : \mathbb{C}^k \to \mathbb{C}^m$ ja $\boldsymbol{\mathcal{N}} : \mathbb{C}^n \to \mathbb{C}^k$ voidaan muodostaa yhdistetty kuvaus

$$\begin{aligned}\boldsymbol{\mathcal{M}} \circ \boldsymbol{\mathcal{N}}(\boldsymbol{z}) &= \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{N}}(\boldsymbol{z})) = \boldsymbol{M}(\boldsymbol{N}\boldsymbol{z} + \boldsymbol{N}_{\#}\bar{\boldsymbol{z}}) + \boldsymbol{M}_{\#}(\overline{\boldsymbol{N}\boldsymbol{z} + \boldsymbol{N}_{\#}\bar{\boldsymbol{z}}}) \\ &= (\boldsymbol{M}\boldsymbol{N} + \boldsymbol{M}_{\#}\overline{\boldsymbol{N}}_{\#})\boldsymbol{z} + (\boldsymbol{M}\boldsymbol{N}_{\#} + \boldsymbol{M}_{\#}\overline{\boldsymbol{N}})\bar{\boldsymbol{z}}.\end{aligned}$$

Tämä on $\mathbb{R}$-lineaarinen ja sitä merkitään myös lyhyemmin $\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}}(\boldsymbol{z}) = \boldsymbol{\mathcal{M}} \circ \boldsymbol{\mathcal{N}}(\boldsymbol{z})$. Kohdan (1.1) mukainen $m \times n$-skalaarioperaattorimuoto $\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}} = (\boldsymbol{\pi}_{ij})$ sille saadaan $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mu}_{ij})$:n ja $\boldsymbol{\mathcal{N}} = (\boldsymbol{\nu}_{ij})$:n muodoista laskemalla

$$\boldsymbol{\pi}_{ij}(z) = \sum_{l=1}^{k} \boldsymbol{\mu}_{il}(\boldsymbol{\nu}_{lj}(z)) = \sum_{l=1}^{k} \boldsymbol{\mu}_{il} \circ \boldsymbol{\nu}_{lj}(z) \qquad (1 \le i \le m\,,\ 1 \le j \le n).$$

$\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}}$:ää vastaava reaalinen matriisi saadaan $\boldsymbol{\mathcal{M}}$:ää ja $\boldsymbol{\mathcal{N}}$:ää vastaavien matriisien tulona.

Määritelmä 1.1.4. *Olkoon $\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^n$ reaalilineaarinen operaattori. $\mathbb{R}$-lineaarista operaattoria $\boldsymbol{\mathcal{N}} : \mathbb{C}^n \to \mathbb{C}^n$ kutsutaan $\boldsymbol{\mathcal{M}}$:n käänteisoperaattoriksi, jos*

$$\boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{N}}(\boldsymbol{z})) = \boldsymbol{z} \qquad \textit{ja} \qquad \boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{M}}(\boldsymbol{z})) = \boldsymbol{z} \qquad \textit{kaikilla } \boldsymbol{z} \in \mathbb{C}^n.$$

Käänteisoperaattoria merkitään $\boldsymbol{\mathcal{M}}^{-1} = \boldsymbol{\mathcal{N}}$. Toisin sanoen, $\boldsymbol{\mathcal{N}}$ on $\boldsymbol{\mathcal{M}}$:n käänteisoperaattori, jos $\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}} = \boldsymbol{I}$ ja $\boldsymbol{\mathcal{N}}\boldsymbol{\mathcal{M}} = \boldsymbol{I}$, missä $\boldsymbol{I}$ on yksikköoperaattori (yksikkömatriisi) $\boldsymbol{I}\boldsymbol{z} = \boldsymbol{z}$ kaikilla $\boldsymbol{z} \in \mathbb{C}^n$.

$\boldsymbol{\mathcal{M}}$:n käänteisoperaattori $\boldsymbol{\mathcal{M}}^{-1}$ on olemassa täsmälleen silloin kun sitä vastaavalla reaalisella (esim. tavoin (1.3) tai (1.4) muodostetulla) $2n \times 2n$-matriisilla on käänteismatriisi. Jos tällaiselle reaaliselle matriisille muodostetaan käänteismatriisi, vastaa se operaattoria $\boldsymbol{\mathcal{M}}^{-1}$. Reaalimatriisisamaistuksen perusteella voidaan myös nähdä, että $\mathbb{R}$-operaattorilla $\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^n$ on olemassa käänteisoperaattori mikäli $\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}} = \boldsymbol{I}$ *tai* $\boldsymbol{\mathcal{N}}\boldsymbol{\mathcal{M}} = \boldsymbol{I}$ jollakin operaattorilla $\boldsymbol{\mathcal{N}}$; toista ehdoista ei siis välttämättä tarvita.

Käänteisoperaattori on yksikäsitteinen, kuten $2n \times 2n$-reaalimatriisisamaistuksen perusteella voidaan havaita. Voidaan todeta myös suoraan: mikäli $\boldsymbol{\mathcal{N}}_1$ ja $\boldsymbol{\mathcal{N}}_2$ ovat kaksi käänteisoperaattoria, niin $\boldsymbol{\mathcal{N}}_1 = \boldsymbol{\mathcal{N}}_1\boldsymbol{I} = \boldsymbol{\mathcal{N}}_1(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}}_2) = (\boldsymbol{\mathcal{N}}_1\boldsymbol{\mathcal{M}})\boldsymbol{\mathcal{N}}_2 = \boldsymbol{I}\boldsymbol{\mathcal{N}}_2 = \boldsymbol{\mathcal{N}}_2$.

Esimerkki 1.1.5. Olkoon $\boldsymbol{\mu} : \mathbb{C} \to \mathbb{C}$ reaalilineaarinen skalaarioperaattori, $\boldsymbol{\mu}(z) = uz + v\bar{z}$, missä $u, v \in \mathbb{C}$. Tapaa (1.3) vastaava reaalinen matriisi on tällöin

$$\begin{bmatrix} \mathrm{Re}(u) + \mathrm{Re}(v) & -\mathrm{Im}(u) + \mathrm{Im}(v) \\ \mathrm{Im}(u) + \mathrm{Im}(v) & \mathrm{Re}(u) - \mathrm{Re}(v) \end{bmatrix}.$$

Kun tälle lasketaan käänteismatriisi saadaan

$$\frac{1}{\mathrm{Re}(u)^2 + \mathrm{Im}(u)^2 - \mathrm{Re}(v)^2 - \mathrm{Im}(v)^2} \begin{bmatrix} \mathrm{Re}(u) + \mathrm{Re}(-v) & -(-\mathrm{Im}(u)) + \mathrm{Im}(-v) \\ -\mathrm{Im}(u) + \mathrm{Im}(-v) & \mathrm{Re}(u) - \mathrm{Re}(-v) \end{bmatrix}.$$

Lukemalla tätä ja muotoa (1.3) takaperin, tulee käänteisoperaattori muotoon

$$\boldsymbol{\mu}^{-1}(z) = \frac{1}{|u|^2 - |v|^2}(\bar{u}z - v\bar{z}).$$

Nimittäjässä oleva luku $|u|^2-|v|^2$ on matriisin determinantti ja siten $\boldsymbol{\mu}$:n käänteisoperaattori on olemassa täsmälleen silloin, kun $|u| \neq |v|$. Voitaisiin myös etsiä suoraan operaattoria $\boldsymbol{\nu} : \mathbb{C} \to \mathbb{C}$ siten, että $\boldsymbol{\nu\mu}(z) = z$ kaikilla $z \in \mathbb{C}$. Olkoon $\boldsymbol{\nu}(z) = pz+q\bar{z}$ missä $p, q \in \mathbb{C}$. Laskemalla saadaan $\boldsymbol{\nu\mu}(z) = (pu+q\bar{v})z+(pv+q\bar{u})\bar{z}$. Yhtälöparista $pu+q\bar{v} = 1$, $pv+q\bar{u} = 0$ saadaan ratkaistua $p = \bar{u}/(|u|^2-|v|^2)$, $q = -v/(|u|^2-|v|^2)$.

1.1.3 Liitto-operaattori

Kun kohdan 1.3 matriisi transponoidaan, on tuloksena

$$\begin{bmatrix} \mathrm{Re}(\boldsymbol{M}^T)+\mathrm{Re}(\boldsymbol{M}_{\#}^T) & -(-\mathrm{Im}(\boldsymbol{M}^T))+\mathrm{Im}(\boldsymbol{M}_{\#}^T) \\ -\mathrm{Im}(\boldsymbol{M}^T)+\mathrm{Im}(\boldsymbol{M}_{\#}^T) & \mathrm{Re}(\boldsymbol{M}^T)-\mathrm{Re}(\boldsymbol{M}_{\#}^T) \end{bmatrix}.$$

Kulkemalla reaalimatriisiesityksestä takaisin operaattoriksi, saadaan

$$\boldsymbol{\mathcal{M}}^*(\boldsymbol{z}) = \boldsymbol{M}^*\boldsymbol{z}+\boldsymbol{M}_{\#}^T\bar{\boldsymbol{z}}$$

ja sitä kutsutaan operaattorin $\boldsymbol{\mathcal{M}}$ *liitto-operaattoriksi.* Täten erityisesti skalaarioperaattorin $\boldsymbol{\mu}(z) = uz+v\bar{z}$ liitto-operaattori on $\boldsymbol{\mu}^*(z) = \bar{u}z+v\bar{z}$, jolloin skalaarimuodossa (1.1) ilmaisten

$$\boldsymbol{\mathcal{M}}^* = \begin{bmatrix} \boldsymbol{\mu}_{11}^* & \boldsymbol{\mu}_{21}^* & \boldsymbol{\mu}_{31}^* & \cdots & \boldsymbol{\mu}_{m1}^* \\ \boldsymbol{\mu}_{12}^* & \boldsymbol{\mu}_{22}^* & \boldsymbol{\mu}_{32}^* & \cdots & \boldsymbol{\mu}_{m2}^* \\ \boldsymbol{\mu}_{13}^* & \boldsymbol{\mu}_{23}^* & \boldsymbol{\mu}_{33}^* & \cdots & \boldsymbol{\mu}_{m3}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mu}_{1n}^* & \boldsymbol{\mu}_{2n}^* & \boldsymbol{\mu}_{3n}^* & \cdots & \boldsymbol{\mu}_{mn}^* \end{bmatrix}.$$

Kahden operaattorin yhdisteelle pätee $(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}})^* = \boldsymbol{\mathcal{N}}^*\boldsymbol{\mathcal{M}}^*$. Liitto-operaattorista hieman poiketen, operaattorin $\boldsymbol{\mathcal{M}}$ *transpoosi* on

$$\boldsymbol{\mathcal{M}}^T(\boldsymbol{z}) = \boldsymbol{M}^T\boldsymbol{z}+\boldsymbol{M}_{\#}^T\bar{\boldsymbol{z}}.$$

Tällöin

$$\boldsymbol{\mathcal{M}}^T = \begin{bmatrix} \boldsymbol{\mu}_{11} & \boldsymbol{\mu}_{21} & \boldsymbol{\mu}_{31} & \cdots & \boldsymbol{\mu}_{m1} \\ \boldsymbol{\mu}_{12} & \boldsymbol{\mu}_{22} & \boldsymbol{\mu}_{32} & \cdots & \boldsymbol{\mu}_{m2} \\ \boldsymbol{\mu}_{13} & \boldsymbol{\mu}_{23} & \boldsymbol{\mu}_{33} & \cdots & \boldsymbol{\mu}_{m3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mu}_{1n} & \boldsymbol{\mu}_{2n} & \boldsymbol{\mu}_{3n} & \cdots & \boldsymbol{\mu}_{mn} \end{bmatrix}.$$

Yleensä kahden operaattorin yhdisteelle $(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{N}})^T \neq \boldsymbol{\mathcal{N}}^T\boldsymbol{\mathcal{M}}^T$.

Määritelmä 1.1.6. *Reaalilineaarinen operaattori $\boldsymbol{\mathcal{Q}} : \mathbb{C}^n \to \mathbb{C}^m$ on isometria, jos*

$$\|\boldsymbol{\mathcal{Q}}(\boldsymbol{z})\| = \|\boldsymbol{z}\| \qquad \textit{kaikilla } \boldsymbol{z} \in \mathbb{C}^n.$$

Sitä kutsutaan unitaariseksi, jos $m = n$.

Lause 1.1.7. *Reaalilineaarinen operaattori $\boldsymbol{\mathcal{U}} : \mathbb{C}^n \to \mathbb{C}^n$ on unitaarinen jos ja vain jos $\boldsymbol{\mathcal{U}}^*\boldsymbol{\mathcal{U}} = \boldsymbol{I}$.*

Todistus. Olkoon matriisi $\boldsymbol{A} \in \mathbb{R}^{2n\times 2n}$ unitaarioperaattorin $\boldsymbol{\mathcal{U}} : \mathbb{C}^n \to \mathbb{C}^n$ reaalimatriisiesitys. Tällöin $\|\boldsymbol{Ax}\| = \|\boldsymbol{x}\|$ kaikilla $\boldsymbol{x} \in \mathbb{R}^{2n}$. Kun merkitään avaruuden $\mathbb{R}^{2n}$ sisätuloa $(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{y}$, niin kaikilla $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{2n}$

$$\begin{aligned}\|\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{y})\|^2 = \|\boldsymbol{x}+\boldsymbol{y}\|^2 &\Leftrightarrow \big(\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{y}), \boldsymbol{A}(\boldsymbol{x}+\boldsymbol{y})\big) = (\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x}+\boldsymbol{y}) \\ &\Leftrightarrow (\boldsymbol{Ax}, \boldsymbol{Ay}) = (\boldsymbol{x}, \boldsymbol{y}) \Leftrightarrow (\boldsymbol{A}^T\boldsymbol{Ax}, \boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{y}) \\ &\Leftrightarrow (\boldsymbol{A}^T\boldsymbol{Ax} - \boldsymbol{x}, \boldsymbol{y}) = 0.\end{aligned}$$

Valitsemalla $\boldsymbol{y} = \boldsymbol{A}^T\boldsymbol{Ax} - \boldsymbol{x}$, todetaan $\boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{x}$ kaikilla $\boldsymbol{x} \in \mathbb{R}^{2n}$. Täten $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$.

Toisaalta, jos $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$, niin kaikilla $\boldsymbol{x}$

$$(\boldsymbol{A}^T\boldsymbol{Ax}, \boldsymbol{x}) = (\boldsymbol{x}, \boldsymbol{x}) \Leftrightarrow (\boldsymbol{Ax}, \boldsymbol{Ax}) = (\boldsymbol{x}, \boldsymbol{x}) \Leftrightarrow \|\boldsymbol{Ax}\|^2 = \|\boldsymbol{x}\|^2.$$

□

1.2 QR-hajotelma

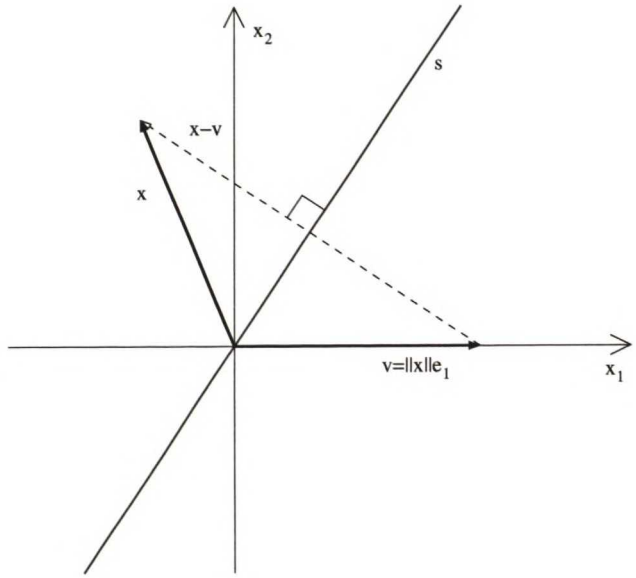
Muun muassa matriisien QR-hajotelman laskemista varten tarvitaan unitaarimatriisia, jolla kertominen kääntää annetun vektorin $\boldsymbol{x}$ luonnollisen kantavektorin $\boldsymbol{e}_1$ suuntaiseksi. Tätä varten tarkastellaan Householderin peilausmatriisia $\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{ww}^*$, missä $\boldsymbol{w} \in \mathbb{C}^n$ on yksikkövektori. Tämä on selvästi hermiittinen ja unitaarinen $\boldsymbol{H}^* = \boldsymbol{H}$, $\boldsymbol{H}^*\boldsymbol{H} = \boldsymbol{H}^2 = \boldsymbol{I}$. Vektori $\boldsymbol{w}$ on peilaavan tason normaalivektori ja lausekkeesta $\boldsymbol{Hx} = \boldsymbol{x} - 2\boldsymbol{w}(\boldsymbol{w}^*\boldsymbol{x})$ nähdään, että $\boldsymbol{x}$:n normaalivektorin suuntainen komponentti vähennetään kaksi kertaa. Siten $\boldsymbol{H}$:lla kertominen siirtää $\boldsymbol{x}$:n peilaavan tason toiselle puolelle samaan asemaan.

Kuva 1.1 havainnollistaa reaalisen vektorin $\boldsymbol{x} \in \mathbb{R}^n$ peilaamisen vektoriksi $\boldsymbol{v} = \|\boldsymbol{x}\|\boldsymbol{e}_1$. Peilaavaa tasoa esittää suora s ja nähdään, että normaalivektori on $\boldsymbol{w} = (\boldsymbol{x}-\boldsymbol{v})/\|\boldsymbol{x}-\boldsymbol{v}\|$. Kun vektori $\boldsymbol{x}$ on lähes $\boldsymbol{e}_1$:den suuntainen, on tässä kaavassa nimittäjä lähellä nollaa. Numeerisessa laskennassa kannattaakin tällöin peilata $\boldsymbol{x}$ vektorin $-\boldsymbol{e}_1$ suuntaan. Valinta voidaan perustaa $\boldsymbol{x}$:n ensimmäisen komponentin etumerkkiin, jolloin saadaan kaava

$$\boldsymbol{w} = \frac{\boldsymbol{x} + \alpha\|\boldsymbol{x}\|\boldsymbol{e}_1}{\big\|\boldsymbol{x} + \alpha\|\boldsymbol{x}\|\boldsymbol{e}_1\big\|}, \qquad \alpha = \begin{cases} \frac{\langle \boldsymbol{x}, \boldsymbol{e}_1\rangle}{|\langle \boldsymbol{x}, \boldsymbol{e}_1\rangle|}, & \text{kun } \langle \boldsymbol{x}, \boldsymbol{e}_1\rangle \neq 0, \\ 1, & \text{kun } \langle \boldsymbol{x}, \boldsymbol{e}_1\rangle = 0. \end{cases} \tag{1.5}$$

Kompleksivektorin $\boldsymbol{x} \in \mathbb{C}^n$ tapauksessa ei voida luottaa geometriseen intuitioon ja joudutaan laskemaan. Yritetään peilata $\boldsymbol{x}$ vektoriksi $\boldsymbol{v}$, jolle $\|\boldsymbol{v}\| = \|\boldsymbol{x}\|$ ja $\boldsymbol{v} \neq \boldsymbol{x}$. Kokeillaan yllä olevan perusteella $\boldsymbol{w} = (\boldsymbol{x} - \boldsymbol{v})/\|\boldsymbol{x} - \boldsymbol{v}\|$, jolloin

$$\begin{aligned}\boldsymbol{Hx} &= \boldsymbol{x} - 2\frac{(\boldsymbol{x}-\boldsymbol{v})(\boldsymbol{x}-\boldsymbol{v})^*}{\|\boldsymbol{x}-\boldsymbol{v}\|^2}\boldsymbol{x} = \boldsymbol{v} + (\boldsymbol{x}-\boldsymbol{v})(\boldsymbol{x}-\boldsymbol{v})^*\frac{(\boldsymbol{x}-\boldsymbol{v}) - 2\boldsymbol{x}}{\|\boldsymbol{x}-\boldsymbol{v}\|^2} \\ &= \boldsymbol{v} - (\boldsymbol{x}-\boldsymbol{v})\frac{(\boldsymbol{x}^*-\boldsymbol{v}^*)(\boldsymbol{x}+\boldsymbol{v})}{\|\boldsymbol{x}-\boldsymbol{v}\|^2} = \boldsymbol{v} - (\boldsymbol{x}-\boldsymbol{v})\frac{\|\boldsymbol{x}\|^2 + \boldsymbol{x}^*\boldsymbol{v} - \boldsymbol{v}^*\boldsymbol{x} - \|\boldsymbol{v}\|^2}{\|\boldsymbol{x}-\boldsymbol{v}\|^2} \\ &= \boldsymbol{v} + (\boldsymbol{x}-\boldsymbol{v})\frac{2i\,\mathrm{Im}(\boldsymbol{v}^*\boldsymbol{x})}{\|\boldsymbol{x}-\boldsymbol{v}\|^2}.\end{aligned}$$



Kuva 1.1: Vektorin $\boldsymbol{x}$ heijastaminen luonnollisen kantavektorin $\boldsymbol{e}_1$ suuntaiseksi. Suora s esittää heijastavaa tasoa.

Täytyy siis lisäksi vaatia $\operatorname{Im}\langle \boldsymbol{x}, \boldsymbol{v}\rangle = 0$. Näin onkin, kun $\boldsymbol{v} = -\alpha\|\boldsymbol{x}\|\boldsymbol{e}_1$, missä α on kaavasta (1.5). Täten kaava (1.5) on pätevä myös kompleksivektoreille. Kun vektori $\boldsymbol{x} \in \mathbb{C}^n$ on annettu, laskemalla $\boldsymbol{w}$ tästä kaavasta, pätee $\boldsymbol{H}\boldsymbol{x} = -\alpha\|\boldsymbol{x}\|\boldsymbol{e}_1$.

Siirrytään sitten matriiseista reaalilineaarisiin operaattoreihin. Tällöin tarvitaan unitaarioperaattoria, jolla operointi kääntää *kaksi* annettua vektoria kantavektorin $\boldsymbol{e}_1$ suuntaiseksi. Olkoot annetut vektorit $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$ ja oletetaan ne lineaarisesti riippumattomiksi. Lineaarisesti riippuville $\boldsymbol{x}, \boldsymbol{y}$ voidaan käyttää edellä kuvattua tavallista Householderin muunnosta.

Operaattoria $\mathcal{H}(\boldsymbol{z}) = \boldsymbol{z} - \boldsymbol{U}\boldsymbol{U}^*\boldsymbol{z} - \boldsymbol{U}\boldsymbol{U}^T\bar{\boldsymbol{z}}$, missä $\boldsymbol{U} \in \mathbb{C}^{n\times 2}$ ja $\operatorname{Re}(\boldsymbol{U}^*\boldsymbol{U}) = \boldsymbol{I}$ kutsutaan reaalilineaariseksi Householderin muunnokseksi. Laskemalla nähdään, että $\mathcal{H}^* = \mathcal{H}$ ja $\mathcal{H}^2 = \boldsymbol{I}$, joten erityisesti $\mathcal{H}$ on unitaarinen. Lähdetaan seuraavaksi etsimään matriisia $\boldsymbol{U}$, jolla $\boldsymbol{x}$ ja $\boldsymbol{y}$ kääntyvät vektorin $\boldsymbol{e}_1$ suuntaan. Merkitään $\boldsymbol{V} = \begin{bmatrix}\boldsymbol{x} & \boldsymbol{y}\end{bmatrix}$, $\boldsymbol{e} = \boldsymbol{e}_1$ ja $\boldsymbol{a} = \begin{bmatrix}a_1 & a_2\end{bmatrix}^T \in \mathbb{C}^{2\times 1}$, jolloin ehdot $\mathcal{H}(\boldsymbol{x}) = a_1\boldsymbol{e}$ ja $\mathcal{H}(\boldsymbol{y}) = a_2\boldsymbol{e}$ voidaan kirjoittaa yhtäpitävästi matriisimuotoon

$$\boldsymbol{V} - \boldsymbol{U}\boldsymbol{U}^*\boldsymbol{V} - \boldsymbol{U}\boldsymbol{U}^T\overline{\boldsymbol{V}} = \boldsymbol{V} - 2\boldsymbol{U}\operatorname{Re}(\boldsymbol{U}^*\boldsymbol{V}) = \boldsymbol{e}\boldsymbol{a}^*. \tag{1.6}$$

Jotta ratkaisu $\boldsymbol{U}$ voisi olla olemassa, täytyy jollakin matriisilla $\boldsymbol{R} \in \mathbb{R}^{2\times 2}$ olla $\boldsymbol{U}\boldsymbol{R} = (\boldsymbol{V} - \boldsymbol{e}\boldsymbol{a}^*)$. Tällöin

$$\begin{aligned}\boldsymbol{V} - 2\boldsymbol{U}\operatorname{Re}(\boldsymbol{U}^*\boldsymbol{V}) &= \boldsymbol{e}\boldsymbol{a}^* + \boldsymbol{U}\boldsymbol{R} - 2\boldsymbol{U}\operatorname{Re}(\boldsymbol{U}^*\boldsymbol{V}) = \boldsymbol{e}\boldsymbol{a}^* + \boldsymbol{U}\operatorname{Re}(\boldsymbol{R} - 2\boldsymbol{U}^*\boldsymbol{V}) \\ &= \boldsymbol{e}\boldsymbol{a}^* + \boldsymbol{U}\operatorname{Re}(\boldsymbol{U}^*\boldsymbol{U}\boldsymbol{R} - 2\boldsymbol{U}^*\boldsymbol{V}) = \boldsymbol{e}\boldsymbol{a}^* - \boldsymbol{U}\operatorname{Re}(\boldsymbol{U}^*(\boldsymbol{V} + \boldsymbol{e}\boldsymbol{a}^*)).\end{aligned} \tag{1.7}$$

Jos löydetään $\boldsymbol{U}$, kääntyvä $\boldsymbol{R}$ ja vektori $\boldsymbol{a}$ siten, että

$$\boldsymbol{U}\boldsymbol{R} = \boldsymbol{V} - \boldsymbol{e}\boldsymbol{a}^* \qquad \text{ja} \qquad \operatorname{Re}((\boldsymbol{V} - \boldsymbol{e}\boldsymbol{a}^*)^*(\boldsymbol{V} + \boldsymbol{e}\boldsymbol{a}^*)) = 0, \tag{1.8}$$

niin $\operatorname{Re}(\boldsymbol{U}^*(\boldsymbol{V} + \boldsymbol{e}\boldsymbol{a}^*)) = (\boldsymbol{R}^{-1})^*\operatorname{Re}((\boldsymbol{V} - \boldsymbol{e}\boldsymbol{a}^*)^*(\boldsymbol{V} + \boldsymbol{e}\boldsymbol{a}^*)) = 0$, joten kaavan (1.7) mukaan $\boldsymbol{U}$ on sopiva ratkaisu. Ryhdytään sitten etsimään ehdot (1.8) toteuttavia $\boldsymbol{U}$,

$\boldsymbol{R}$ ja $\boldsymbol{a}$. Merkitään $\boldsymbol{w} = \boldsymbol{V}^*\boldsymbol{e}$, jolloin jälkimmäinen ehto

$$\mathrm{Re}(\boldsymbol{V}^*\boldsymbol{V} + \boldsymbol{w}\boldsymbol{a}^* - \boldsymbol{a}\boldsymbol{w}^* - \boldsymbol{a}\boldsymbol{a}^*) = 0.$$

Merkitään $\boldsymbol{c} = \mathrm{Re}(\boldsymbol{V}^*\boldsymbol{V})^{1/2}\left[\begin{smallmatrix}1\\ i\end{smallmatrix}\right]$, jolloin $\mathrm{Re}(\boldsymbol{V}^*\boldsymbol{V} - \boldsymbol{c}\boldsymbol{c}^*) = 0$. Kun $\boldsymbol{a} = \eta\boldsymbol{c}$, missä $\eta \in \mathbb{C}$, $|\eta| = 1$, niin myös $\mathrm{Re}(\boldsymbol{V}^*\boldsymbol{V} - \boldsymbol{a}\boldsymbol{a}^*) = 0$. Etsitään vielä η siten, että $\mathrm{Re}(\boldsymbol{w}\boldsymbol{a}^* - \boldsymbol{a}\boldsymbol{w}^*) = 0$, mikä yhtäpitävästi kirjoitettuna on $\mathrm{Re}(w_1\overline{a}_2 - a_1\overline{w}_2) = 0$ ja

$$\mathrm{Re}(w_1\overline{\eta c_2} - \eta c_1\overline{w}_2) = 0.$$

Edelleen yhtäpitävästi tämä on $\eta q + \overline{\eta q} = 0$, missä $q = \overline{w_1}c_2 - \overline{w_2}c_1$. Kun $q \neq 0$, niin ratkaisut ovat $\eta = \pm i\frac{\overline{q}}{|q|}$. Seuraavan helposti todistettavan lemman mukaan ainakin toisella näistä matriisin $(\boldsymbol{V} - \boldsymbol{e}\boldsymbol{a}^*)$ sarakkeet ovat lineaarisesti riippumattomat.

Lemma 1.2.1. *Olkoot $\boldsymbol{x}$ ja $\boldsymbol{y}$ lineaarisesti riippumattomia vektoreita. Kun $\boldsymbol{z}$ on vektori ja α_1, α_2 skalaareita, niin ainakin toinen joukoista $\{\boldsymbol{x} - \alpha_1\boldsymbol{z}, \boldsymbol{y} - \alpha_2\boldsymbol{z}\}$ ja $\{\boldsymbol{x} + \alpha_1\boldsymbol{z}, \boldsymbol{y} + \alpha_2\boldsymbol{z}\}$ on lineaarisesti riippumaton.*

Lopuksi vielä ortonormalisoidaan matriisin $(\boldsymbol{V} - \boldsymbol{e}\boldsymbol{a}^*)$ sarakkeet reaalisen sisätulon $(\boldsymbol{u}, \boldsymbol{v}) = \mathrm{Re}(\boldsymbol{v}^*\boldsymbol{u})$ suhteen. Yhteenvetona edellä oleva voidaan kirjoittaa seuraaviksi kaavoiksi $\boldsymbol{U}$:n laskemiseksi.

$$\begin{aligned}
\boldsymbol{c} &= \mathrm{Re}(\boldsymbol{V}^*\boldsymbol{V})^{1/2}\left[\begin{smallmatrix}1\\ i\end{smallmatrix}\right], \quad q = \langle \boldsymbol{V}\boldsymbol{J}\boldsymbol{c}, \boldsymbol{e}\rangle \quad (\boldsymbol{J} = \left[\begin{smallmatrix}0 & -1\\ 1 & 0\end{smallmatrix}\right]),\\
\alpha &= \begin{cases} \frac{q}{|q|}, & \text{kun } q \neq 0,\\ 1, & \text{kun } q = 0. \end{cases}\\
\boldsymbol{W}_- &= \boldsymbol{V} - i\alpha\boldsymbol{e}\boldsymbol{c}^*, \quad \boldsymbol{W}_+ = \boldsymbol{V} + i\alpha\boldsymbol{e}\boldsymbol{c}^*,\\
\boldsymbol{W} &= \begin{cases} \boldsymbol{W}_+, & \text{kun } \det(\mathrm{Re}(\boldsymbol{W}_+^*\boldsymbol{W}_+)) \geq \det(\mathrm{Re}(\boldsymbol{W}_-^*\boldsymbol{W}_-)),\\ \boldsymbol{W}_-, & \text{kun } \det(\mathrm{Re}(\boldsymbol{W}_+^*\boldsymbol{W}_+)) < \det(\mathrm{Re}(\boldsymbol{W}_-^*\boldsymbol{W}_-)). \end{cases}\\
\boldsymbol{Q}\boldsymbol{R} &= \begin{bmatrix}\mathrm{Re}(\boldsymbol{W})\\ \mathrm{Im}(\boldsymbol{W})\end{bmatrix} \qquad \text{(lasketaan suppea QR-hajotelma)},\\
\boldsymbol{U} &= \begin{bmatrix}\boldsymbol{I}_n & 0\end{bmatrix}\boldsymbol{Q} + i\begin{bmatrix}0 & \boldsymbol{I}_n\end{bmatrix}\boldsymbol{Q}.
\end{aligned}$$

Huomautus 1.2.2. Matriisin $\boldsymbol{A}$ sarakkeet ovat lineaarisesti riippumattomat $\mathbb{C}$:n yli, jos ja vain jos $\boldsymbol{A}^*\boldsymbol{A}$ on positiividefiniitti. Sarakkeet ovat lineaarisesti riippumattomat $\mathbb{R}$:n yli, jos ja vain jos $\mathrm{Re}(\boldsymbol{A}^*\boldsymbol{A})$ on positiividefiniitti.

Huomautus 1.2.3. Valinta $\boldsymbol{W} = \boldsymbol{W}_+$ vaikuttaa numeeristen kokeiden perusteella toteutuvan aina ja on analoginen valinta kaavan (1.5) kanssa. Todistus tälle seikalle kuitenkin puuttuu.

Reaalilineaarisen operaattorin $\mathcal{M}$ QR-hajotelman muodostamista varten tarvitaan muunnoksia, joilla muodossa (1.1) ilmaistun operaattorin sarakkeita saadaan ensimmäistä alkiota lukuunottamatta nollaksi. Toisin sanoen, jos $\boldsymbol{\mu}_i(z) = u_i z + v_i\bar{z}\,(i = 1, 2, \ldots, n)$ ovat skalaarioperaattoreita, niin halutaan löytää reaalilineaarinen Householderin muunnos, jolla

$$\mathcal{H} \circ \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\\ \vdots\\ \boldsymbol{\mu}_n\end{bmatrix} = \begin{bmatrix}\#\\ 0\\ \vdots\\ 0\end{bmatrix}. \tag{1.9}$$

Merkitään $\boldsymbol{u} = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}^T$ ja $\boldsymbol{v} = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}^T$, jolloin

$$\begin{aligned} \mathcal{H}(\boldsymbol{u}z + \boldsymbol{v}\bar{z}) &= \mathcal{H}(\boldsymbol{u}z) + \mathcal{H}(\boldsymbol{v}\bar{z}) \\ &= \mathrm{Re}(z)\mathcal{H}(\boldsymbol{u}) + \mathrm{Im}(z)\mathcal{H}(i\boldsymbol{u}) + \mathrm{Re}(z)\mathcal{H}(\boldsymbol{v}) + \mathrm{Im}(z)\mathcal{H}(-i\boldsymbol{v}) \\ &= \mathrm{Re}(z)\mathcal{H}(\boldsymbol{u} + \boldsymbol{v}) + \mathrm{Im}(z)\mathcal{H}\big(i(\boldsymbol{u} - \boldsymbol{v})\big). \end{aligned}$$

Ehto (1.9) siis toteutuu, kun $\mathcal{H}$:ksi valitaan Householderin muunnos, jolla operointi kääntää vektorit $(\boldsymbol{u} + \boldsymbol{v})$ ja $i(\boldsymbol{u} - \boldsymbol{v})$ kantavektorin $\boldsymbol{e}_1$ suuntaisiksi.

Lause 1.2.4 (QR-hajotelma). *Olkoon $\mathcal{M} : \mathbb{C}^n \to \mathbb{C}^m$ reaalilineaarinen operaattori ja $m \geq n$. Tällöin on olemassa unitaarinen $\mathcal{Q} : \mathbb{C}^m \to \mathbb{C}^m$ ja yläkolmio-operaattori $\mathcal{R} : \mathbb{C}^n \to \mathbb{C}^m$ siten, että*

$$\mathcal{M} = \mathcal{Q}\mathcal{R}.$$

Todistus. Olkoon operaattori $\mathcal{M}$ ilmaistu muodossa (1.1), missä $m \geq n$. Valitaan Householderin muunnos $\mathcal{H}_1 : \mathbb{C}^m \to \mathbb{C}^m$ siten, että $\mathcal{M}$:n ensimmäiselle sarakkeelle pätee (1.9). Tällöin

$$\mathcal{H}_1\mathcal{M} = \begin{bmatrix} \boldsymbol{\mu}_{11}^{(1)} & \boldsymbol{\mu}_{12}^{(1)} & \boldsymbol{\mu}_{13}^{(1)} & \cdots & \boldsymbol{\mu}_{1n}^{(1)} \\ 0 & \boldsymbol{\mu}_{22}^{(1)} & \boldsymbol{\mu}_{23}^{(1)} & \cdots & \boldsymbol{\mu}_{2n}^{(1)} \\ 0 & \boldsymbol{\mu}_{32}^{(1)} & \boldsymbol{\mu}_{33}^{(1)} & \cdots & \boldsymbol{\mu}_{3n}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \boldsymbol{\mu}_{m2}^{(1)} & \boldsymbol{\mu}_{m3}^{(1)} & \cdots & \boldsymbol{\mu}_{mn}^{(1)} \end{bmatrix}.$$

Valitaan seuraavaksi muunnos $\widetilde{\mathcal{H}}_2 : \mathbb{C}^{m-1} \to \mathbb{C}^{m-1}$ siten, että (1.9) pätee toisen sarakkeen $(m-1)$:lle jälkimmäiselle alkiolle $\begin{bmatrix} \boldsymbol{\mu}_{22}^{(1)} & \boldsymbol{\mu}_{32}^{(1)} & \cdots & \boldsymbol{\mu}_{m2}^{(1)} \end{bmatrix}^T$. Asetetaan $\mathcal{H}_2 = \begin{bmatrix} \mathbf{1} & 0 \\ 0 & \widetilde{\mathcal{H}}_2 \end{bmatrix}$, missä $\mathbf{1}$ on yksikköskalaarioperaattori $\mathbf{1}(z) = z$. Saadaan

$$\mathcal{H}_2\mathcal{H}_1\mathcal{M} = \begin{bmatrix} \boldsymbol{\mu}_{11}^{(1)} & \boldsymbol{\mu}_{12}^{(1)} & \boldsymbol{\mu}_{13}^{(1)} & \cdots & \boldsymbol{\mu}_{1n}^{(1)} \\ 0 & \boldsymbol{\mu}_{22}^{(2)} & \boldsymbol{\mu}_{23}^{(2)} & \cdots & \boldsymbol{\mu}_{2n}^{(2)} \\ 0 & 0 & \boldsymbol{\mu}_{33}^{(2)} & \cdots & \boldsymbol{\mu}_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \boldsymbol{\mu}_{m3}^{(2)} & \cdots & \boldsymbol{\mu}_{mn}^{(2)} \end{bmatrix}.$$

Jatketaan tällä tavoin kunnes päädytään yläkolmiomuotoon

$$\mathcal{H}_n \cdots \mathcal{H}_2\mathcal{H}_1\mathcal{M} = \begin{bmatrix} \boldsymbol{\mu}_{11}^{(1)} & \boldsymbol{\mu}_{12}^{(1)} & \boldsymbol{\mu}_{13}^{(1)} & \cdots & \boldsymbol{\mu}_{1n}^{(1)} \\ 0 & \boldsymbol{\mu}_{22}^{(2)} & \boldsymbol{\mu}_{23}^{(2)} & \cdots & \boldsymbol{\mu}_{2n}^{(2)} \\ 0 & 0 & \boldsymbol{\mu}_{33}^{(3)} & \cdots & \boldsymbol{\mu}_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \boldsymbol{\mu}_{nn}^{(n)} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Merkitään oikeanpuolimmaista operaattoria $\mathcal{R}$:llä ja asetetaan $\mathcal{Q} = \mathcal{H}_1^*\mathcal{H}_2^* \cdots \mathcal{H}_n^*$, jolloin $\mathcal{M} = \mathcal{Q}\mathcal{R}$. □

Luku 2

Suora ratkaiseminen ja LU-hajotelma

Yhtälön suoralla ratkaisumenetelmällä tarkoitetaan sellaista algoritmia, joka ideaalisessa aritmetiikassa tuottaa yhtälön tarkan ratkaisun etukäteen tunnetulla äärellisellä askelmäärällä. Käytännölliset lineaaristen yhtälöiden suorat menetelmät perustuvat Gaussin eliminointiin ja niitä käytetään, kun kerroinmatriisi on tiheä eikä sillä ole erityistä hyödynnettävää rakennetta. Isokokoiset ja suuren määrän nollia sisältävät (ns. harvat) matriisit soveltuvat paremmin seuraavassa luvussa esiteltävin menetelmin ratkaistaviksi. On olemassa myös harvoille matriiseille soveltuvia Gaussin eliminointiin pohjautuvia menetelmiä, mutta niitä ei tässä työssä käsitellä.

Tässä luvussa esitellään suoria ratkaisumenetelmiä reaalilineaarisille yhtälöryhmille perustuen Gaussin eliminointiin.

2.1 $\mathbb{R}$-lineaaristen operaattoreiden LU-hajotelma

Olkoon $\mathcal{M} : \mathbb{C}^n \to \mathbb{C}^n$ reaalilineaarinen operaattori ja merkitään $\mathcal{M} = (\mu_{ij})_{i,j=1}^n$, missä $\mu_{ij} : \mathbb{C} \to \mathbb{C}$ ovat reaalilineaarisia skaalarioperaattoreita tai samaistuksen kautta reaalisia 2×2-matriiseja, jolloin koko operaattorin $\mathcal{M}$ voidaan ajatellaan olevan reaalinen $2n \times 2n$-matriisi lohkottuna $n \times n$ määrään 2×2 osia. $\mathcal{M}$ oletetaan kääntyväksi tässä luvussa. Tavoitteena on laskea alakolmio-operaattori $\mathcal{L}$, jonka lävistäjällä olevat skalaarioperaattorit ovat yksikköoperaattoreita, ja yläkolmio-operaattori $\mathcal{U}$ siten, että $\mathcal{LU} = \mathcal{M}$.

Kyseistä hajotelmaa kutsutaan, sikäli kun se on olemassa, $\mathcal{M}$:n LU-hajotelmaksi. Sen avulla voidaan ratkaista lineaarinen yhtälöryhmä

$$\mathcal{M}(z) = b, \tag{2.1}$$

missä $b \in \mathbb{C}^n$ on annettu vektori ja ratkaistava vektori on $z \in \mathbb{C}^n$. Jos $\mathcal{M}$:llä on LU-hajotelma, niin tällainen yhtälöryhmä voidaan ratkaista seuraavasti. Ensin ratkaistaan w yhtälöryhmästä $\mathcal{L}(w) = b$ eteenpäin sijoittamalla eli ratkaisemalla w:n alkiot järjestyksessä ensimmäisestä aloittaen. Tämän jälkeen ratkaistaan z yhtälöryhmästä $\mathcal{U}(z) = w$ taaksepäin sijoittamalla eli z:n viimeisestä alkiosta aloittaen.

Ratkaisussa tarvitaan lävistäjällä olevien skalaarioperaattoreiden käänteisoperaattoreita, ks. esimerkki 1.1.5. Tällöin $\mathcal{M}(z) = \mathcal{L}(\mathcal{U}(z)) = \mathcal{L}(w) = b$, joten z ratkaisee yhtälöryhmän (2.1).

LU-hajotelman muodostaminen on kannattavaa esimerkiksi silloin, kun lineaarinen yhtälöryhmä joudutaan ratkaisemaan useaan kertaan eri vektoreilla b, mutta samalla operaattorilla $\mathcal{M}$. Tällöin tarvitaan vähemmän laskutoimituksia, kun ensin muodostetaan $\mathcal{M}$:n LU-hajotelma, kuin ratkaistaessa jokainen yhtälöryhmä erikseen esim. Gaussin eliminoinneilla.

Esitetään seuraavaksi tapa muodostaa LU-hajotelma. Suoritetaan reaalilineaarisia Gaussin eliminaatioita kertomalla operaattori $\mathcal{M} = (\mu_{ij})$ sopivalla operaattorilla $\mathcal{L}_1$ seuraavasti. Olettaen, että $\mathcal{M}$:n ensimmäinen *tukialkio* μ_{11} on kääntyvä saadaan

$$\mathcal{L}_1\mathcal{M} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -\mu_{21}\mu_{11}^{-1} & 1 & 0 & \dots & 0 \\ -\mu_{31}\mu_{11}^{-1} & 0 & 1 & & \\ \vdots & \vdots & & \ddots & \\ -\mu_{n1}\mu_{11}^{-1} & 0 & & & 1 \end{bmatrix} \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \mu_{23} & \cdots & \mu_{2n} \\ \mu_{31} & \mu_{32} & \mu_{33} & \cdots & \mu_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n1} & \mu_{n2} & \mu_{n3} & \cdots & \mu_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1n} \\ 0 & \mu_{22} - \mu_{21}\mu_{11}^{-1}\mu_{12} & \mu_{23} - \mu_{21}\mu_{11}^{-1}\mu_{13} & \cdots & \mu_{2n} - \mu_{21}\mu_{11}^{-1}\mu_{1n} \\ 0 & \mu_{32} - \mu_{31}\mu_{11}^{-1}\mu_{12} & \mu_{33} - \mu_{31}\mu_{11}^{-1}\mu_{13} & \cdots & \mu_{3n} - \mu_{31}\mu_{11}^{-1}\mu_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \mu_{n2} - \mu_{n1}\mu_{11}^{-1}\mu_{12} & \mu_{n3} - \mu_{n1}\mu_{11}^{-1}\mu_{13} & \cdots & \mu_{nn} - \mu_{n1}\mu_{11}^{-1}\mu_{1n} \end{bmatrix} =$$

$$= \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1n} \\ 0 & \nu_{22} & \nu_{23} & \cdots & \nu_{2n} \\ 0 & \nu_{32} & \nu_{33} & \cdots & \nu_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \nu_{n2} & \nu_{n3} & \cdots & \nu_{nn} \end{bmatrix}, \tag{2.2}$$

missä ν_{ij}:llä on merkitty eliminoinnin tuloksena syntyneitä uusia operaattoreita. Jos nyt seuraava $\mathcal{M}$:n tukialkio ν_{22} on kääntyvä, niin eliminointia voidaan jatkaa. Saadaan

$$\mathcal{L}_2\mathcal{L}_1\mathcal{M} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & -\nu_{32}\nu_{22}^{-1} & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & -\nu_{n2}\nu_{22}^{-1} & & & 1 \end{bmatrix} \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1n} \\ 0 & \nu_{22} & \nu_{23} & \cdots & \nu_{2n} \\ 0 & \nu_{32} & \nu_{33} & \cdots & \nu_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \nu_{n2} & \nu_{n3} & \cdots & \nu_{nn} \end{bmatrix} \tag{2.3}$$

$$= \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1n} \\ 0 & \nu_{22} & \nu_{23} & \cdots & \nu_{2n} \\ 0 & 0 & \pi_{33} & \cdots & \pi_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \pi_{n3} & \cdots & \pi_{nn} \end{bmatrix}.$$

Jos kaikki syntyvät $\mathcal{M}$:n tukialkiot ovat kääntyviä, niin tätä toistamalla oikealle puolelle muodostuu lopulta yläkolmio-operaattori; merkitään tätä $\mathcal{U}$:lla. Yllä olevia alakolmio-operaattoreita on merkitty $\mathcal{L}_1$:llä (ensimmäinen näistä) ja $\mathcal{L}_2$:lla (toinen).

Yllä olevan kaltaista prosessia toistettaessa käytettyjä loppuja alakolmio-operaattoreita merkitään $\mathcal{L}_k$:lla. Nämä operaattorit ovat muotoa $\mathcal{L}_k = \mathcal{I} - \eta_k \xi_k^T$, missä η_k on $n \times 1$-operaattori, ξ_k on $n \times 1$-operaattori, jonka k:s komponentti on yksikköoperaattori muiden ollessa nollaoperaattoreita, ja lisäksi pätee $\xi_j^T \eta_k = 0$, kun $j \leq k$. Esimerkiksi $\eta_1 = \begin{bmatrix} 0 & \mu_{21}\mu_{11}^{-1} & \mu_{31}\mu_{11}^{-1} & \cdots & \mu_{n1}\mu_{11}^{-1} \end{bmatrix}^T$.

Edellä olevin merkinnöin

$$\mathcal{L}_{n-1}\mathcal{L}_{n-2}\cdots\mathcal{L}_1\mathcal{M} = \mathcal{U}.$$

Operaattorin $\mathcal{L}_k = \mathcal{I} - \eta_k \xi_k^T$ käänteisoperaattori on $\mathcal{L}_k^{-1} = \mathcal{I} + \eta_k \xi_k^T$. Tämä nähdään laskemalla

$$(\mathcal{I} - \eta_k \xi_k^T)(\mathcal{I} + \eta_k \xi_k^T) = \mathcal{I} + \eta_k \xi_k^T - \eta_k \xi_k^T - \eta_k \xi_k^T \eta_k \xi_k^T = \mathcal{I} - \eta_k (\xi_k^T \eta_k) \xi_k^T = \mathcal{I}.$$

Olkoon $\mathcal{L} = \mathcal{L}_1^{-1} \cdots \mathcal{L}_{n-2}^{-1} \mathcal{L}_{n-1}^{-1}$, joka on alakolmio-operaattori, koska sellaisten tulot säilyvät alakolmio-operaattoreina. Lisäksi $\mathcal{L}$:n lävistäjäalkiot ovat yksikköoperaattoreita. Myös $\mathcal{M} = \mathcal{L}\mathcal{U}$, joten tämä on $\mathcal{M}$:n LU-hajotelma.

Operaattori $\mathcal{L}$ voidaan laskea suoraan Gaussin eliminaatiossa käytetyistä kertoimista. Operaattori $\mathcal{R}_k = \mathcal{L}_k^{-1} \cdots \mathcal{L}_{n-2}^{-1} \mathcal{L}_{n-1}^{-1}$ voidaan kirjoittaa seuraavasti:

$$\mathcal{R}_k = \mathcal{I} + \eta_k \xi_k^T + \cdots + \eta_{n-2} \xi_{n-2}^T + \eta_{n-1} \xi_{n-1}^T.$$

Tämä nähdään induktiolla. Selvästi $\mathcal{R}_{n-1}$ on tätä muotoa. Jos $\mathcal{R}_{k+1}$ on edellä mainittua muotoa, niin

$$\begin{aligned} \mathcal{R}_k &= \mathcal{L}_k^{-1} \mathcal{R}_{k+1} = (\mathcal{I} + \eta_k \xi_k^T)(\mathcal{I} + \eta_{k+1} \xi_{k+1}^T + \cdots + \eta_{n-1} \xi_{n-1}^T) \\ &= \mathcal{I} + \eta_{k+1} \xi_{k+1}^T + \cdots + \eta_{n-1} \xi_{n-1}^T + \eta_k \xi_k^T + \eta_k (\xi_k^T \eta_{k+1}) \xi_{k+1}^T + \cdots + \\ &\quad \eta_k (\xi_k^T \eta_{n-1}) \xi_{n-1}^T = \mathcal{I} + \eta_k \xi_k^T + \eta_{k+1} \xi_{k+1}^T + \cdots + \eta_{n-1} \xi_{n-1}^T, \end{aligned}$$

missä käytettiin ominaisuutta $\xi_j^T \eta_l = 0$, kun $j \leq l$. Siten saadaan

$$\mathcal{L} = \mathcal{R}_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \mu_{21}\mu_{11}^{-1} & 1 & 0 & \cdots & 0 \\ \mu_{31}\mu_{11}^{-1} & \nu_{32}\nu_{22}^{-1} & 1 & & \\ \vdots & \vdots & & \ddots & \\ \mu_{n1}\mu_{11}^{-1} & \nu_{n2}\nu_{22}^{-1} & & & 1 \end{bmatrix}.$$

Operaattoreiden $\mathcal{L}$ ja $\mathcal{U}$ laskemiseksi edellä kuvatulla tavalla täytyy tehdä noin $4n^3/3$ kompleksilukukertolaskua ja saman verran yhteenlaskuja. Tietokoneella laskettaessa kompleksiliukulukulaskutoimituksia (complex flops) tehdään noin $4n^3/3$, koska määritelmän mukaan yksi complex flop tarkoittaa yhden kompleksilukukertolaskun ja -yhteenlaskun yhdistelmää.

Reaalimatriisiesityksestä (1.4) havaitaan, että $\mathbb{R}$-lineaarinen LU-hajotelma voidaan nähdään myös matriisien LU-lohkohajotelmana, missä lohkot ovat 2×2-kokoisia.

2.2 Osittaistuenta

Edellä oletettiin kaikki $\mathcal{M}$:n tukialkiot μ_{11}, ν_{22} jne. kääntyviksi. Tämä ei välttämättä päde, vaikka $\mathcal{M}$ onkin kääntyvä. Lisäksi muihin skalaarioperaattorialkioihin

nähden lähes singulaarinen, mutta kuitenkin kääntyvä, tukialkio voi aiheuttaa äärellistarkkuisessa tietokonearitmetiikassa merkitsevien numeroiden menetyksiä huonontaen lopputuloksen tarkkuutta tai tehden sen jopa hyödyttömäksi. Matriisien LU-hajotelman laskennassa molemmat vastaavat ongelmat voidaan ratkaista käyttämällä tuentamenetelmiä, kuten osittais-, torni- ja täystuenta. Reaalilineaarisille operaattoreille näiden tuentamenetelmien analogioilla ei välttämättä päästä eroon kääntymättömistä tukialkioista, kuten seuraava esimerkki osoittaa.

Esimerkki 2.2.1. Tarkastellaan seuraavaa operaattoria.

$$\boldsymbol{\mu}_{11}(z) = z + \bar{z}, \quad \boldsymbol{\mu}_{12}(z) = iz + i\bar{z}, \quad \boldsymbol{\mu}_{21}(z) = z - \bar{z}, \quad \boldsymbol{\mu}_{22}(z) = -iz + i\bar{z},$$

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \boldsymbol{\mu}_{11} & \boldsymbol{\mu}_{12} \\ \boldsymbol{\mu}_{21} & \boldsymbol{\mu}_{22} \end{bmatrix}.$$

Skalaarioperaattori $\boldsymbol{\mu} : \mathbb{C} \to \mathbb{C}$, $\boldsymbol{\mu}(z) = uz + v\bar{z}$ on kääntyvä täsmälleen silloin, kun $|u| \neq |v|$. Mikään yllä olevista skalaarioperaattoreista ei siis ole kääntyvä. Skalaarioperaattoria $\boldsymbol{\mu}$ vastaava reaalinen matriisi on

$$\begin{bmatrix} \mathrm{Re}(u+v) & -\mathrm{Im}(u-v) \\ \mathrm{Im}(u+v) & \mathrm{Re}(u-v) \end{bmatrix}.$$

$\boldsymbol{\mathcal{M}}$:ää vastaa matriisi

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

Tämä on kääntyvä, joten $\boldsymbol{\mathcal{M}}$ on kääntyvä operaattori. Tarvittavaa kääntyvää tukialkiota ei siis välttämättä ole mahdollista löytää operaattorille $\boldsymbol{\mathcal{M}}$, vaikka $\boldsymbol{\mathcal{M}}$ olisikin kääntyvä. Esimerkin skalaarioperaattoreista $\boldsymbol{\mu}_{ij}$ mikään ei kelpaa ensimmäiseksi tukialkioksi, joten edes matriisien täydellisen tuennan analogia ei toimi kaikilla reaalilineaarisilla operaattoreilla.

Esitellään kuitenkin seuraavaksi matriisien osittaistuennan vastine. Menetelmä laajennetaan myöhemmin ottamaan huomioon myös edellä kuvatun esimerkin kaltaisia tilanteita. Tässä tukialkion valinta suoritetaan rivien järjestystä vaihtamalla. Sarakkeiden järjestyksen vaihtaminen olisi myös mahdollista (saraketuenta) kuten matriisien täydellisen tuennan vastinekin (etsitään jäljellä olevasta alamatriisista itseisarvoltaan suurin alkio ja vaihdetaan sekä rivejä että sarakkeita).

Skalaarioperaattori $\boldsymbol{\mu}(z) = uz + v\bar{z}$ on kääntyvä täsmälleen silloin, kun $|u| \neq |v|$ ja tällöin käänteisoperaattori on

$$\boldsymbol{\mu}^{-1}(z) = \frac{1}{|u|^2 - |v|^2}(\bar{u}z - v\bar{z}).$$

Pyritään välttämään pienellä luvulla jakamista, jottei $\boldsymbol{\mu}^{-1}$:n lineaarisesta eikä antilineaarisesta osasta tulisi itseisarvoltaan suuri. Yritetään siis saada hajotelman laskemisessa käännettävien skalaarioperaattoreiden $|u|^2 - |v|^2$ itseisarvoltaan mahdollisimman suureksi. Merkitään $N(\boldsymbol{\mu}) = |u|^2 - |v|^2$.

Osittaistuennassa etsitään aluksi operaattorin $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mu}_{ij})$ ensimmäiseltä sarakkeelta alkio $\boldsymbol{\mu}_{k,1}$, jolle $|N(\boldsymbol{\mu}_{k,1})|$ on suurin. Vaihdetaan operaattorin $\boldsymbol{\mathcal{M}}$ ensimmäisen ja k:nnen rivin paikkaa. Tämä voidaan tehdä laskemalla $\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$, missä matriisi $\boldsymbol{P}_1 \in \mathbb{C}^{n\times n}$ on permutaatiomatriisi, joka on saatu vaihtamalla yksikkömatriisin

ensimmäisen ja k:nnen sarakkeen paikat. Oletetaan, että löydetylle alkiolle pätee $|N(\boldsymbol{\mu}_{k,1})| > 0$, jolloin operaattorin $\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$ vasemman yläkulman alkio on kääntyvä.

Tämän jälkeen voidaan tehdä Gaussin eliminaatio kuten kaavassa (2.2), mutta nyt $\boldsymbol{\mathcal{M}}$:n paikalla on $\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$. Tämän tuloksena saadaan operaattori $\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$.

Seuraavaksi etsitään matriisin $\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{A}}$ toisen sarakkeen toiselta riviltä alkaen suurinta $|N(\boldsymbol{\mu}_{k,2})|$ eli olkoon $2 \leq k \leq n$ sellainen, että

$$|(\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}})_{k,2}| = \max_{2\leq i\leq n} |(\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}})_{i,2}|.$$

Oletetaan tämä jälleen positiiviseksi. Olkoon $\boldsymbol{P}_2 \in \mathbb{C}^{n\times n}$ permutaatiomatriisi, joka on saatu vaihtamalla yksikkömatriisin sarakkeet 2 ja k keskenään. Gaussin eliminaation tuloksena saadaan $\boldsymbol{\mathcal{L}}_2\boldsymbol{P}_2\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$, kuten kaavassa (2.3), mutta nyt $\boldsymbol{\mathcal{L}}_1\boldsymbol{\mathcal{M}}$:n paikalla on $\boldsymbol{P}_2\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$.

Kun edellä kuvattua menettelyä jatketaan, saadaan yläkolmio-operaattori

$$\boldsymbol{\mathcal{L}}_{n-1}\boldsymbol{P}_{n-1}\boldsymbol{\mathcal{L}}_{n-2}\boldsymbol{P}_{n-2}\cdots\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{U}}.$$

Koska tässä olevat matriisit $\boldsymbol{P}_k$ ovat yksikkömatriiseja mahdollisesti k:s ja jokin sen oikealla puolella oleva sarake vaihdettuna, niin $\boldsymbol{P}_k\boldsymbol{\xi}_j = \boldsymbol{\xi}_j$, kun $j < k$. Myös $\boldsymbol{P}_k = \boldsymbol{P}_k^T$.

Edellisessä kohdassa nähtiin, että $\boldsymbol{\mathcal{L}}_k = \boldsymbol{\mathcal{I}} - \boldsymbol{\eta}_k\boldsymbol{\xi}_k^T$. Matriisit $\boldsymbol{P}_j$ voidaan kuljettaa kaikkien $\boldsymbol{\mathcal{L}}_k$ operaattoreiden perään seuraavalla tavalla. Merkitään $\boldsymbol{\eta}'_{n-2} = \boldsymbol{P}_{n-1}\boldsymbol{\eta}_{n-2}$ ja $\boldsymbol{\mathcal{L}}'_{n-2} = \boldsymbol{\mathcal{I}} - \boldsymbol{\eta}'_{n-2}\boldsymbol{\xi}_{n-2}^T$.

$$\begin{aligned}\boldsymbol{P}_{n-1}\boldsymbol{\mathcal{L}}_{n-2} &= \boldsymbol{P}_{n-1}(\boldsymbol{\mathcal{I}} - \boldsymbol{\eta}_{n-2}\boldsymbol{\xi}_{n-2}^T) = \boldsymbol{P}_{n-1} - \boldsymbol{\eta}'_{n-2}\boldsymbol{\xi}_{n-2}^T \\ &= \boldsymbol{P}_{n-1} - \boldsymbol{\eta}'_{n-2}(\boldsymbol{P}_{n-1}\boldsymbol{\xi}_{n-2})^T = \boldsymbol{P}_{n-1} - \boldsymbol{\eta}'_{n-2}\boldsymbol{\xi}_{n-2}^T\boldsymbol{P}_{n-1}^T \\ &= \boldsymbol{P}_{n-1} - \boldsymbol{\eta}'_{n-2}\boldsymbol{\xi}_{n-2}^T\boldsymbol{P}_{n-1} = \boldsymbol{\mathcal{L}}'_{n-2}\boldsymbol{P}_{n-1}.\end{aligned}$$

Merkitään lisäksi $\boldsymbol{\eta}'_k = \boldsymbol{P}_{n-1}\boldsymbol{P}_{n-2}\cdots\boldsymbol{P}_{k+1}\boldsymbol{\eta}_k$, $\boldsymbol{L}'_k = \boldsymbol{\mathcal{I}} - \boldsymbol{\eta}'_k\boldsymbol{\xi}_k^T$ ja

$$\boldsymbol{P} = \boldsymbol{P}_{n-1}\boldsymbol{P}_{n-2}\cdots\boldsymbol{P}_1.$$

Edellä olevan kaltaisten laskujen jälkeen tuloksena on $\boldsymbol{\mathcal{L}}_{n-1}\boldsymbol{\mathcal{L}}'_{n-2}\cdots\boldsymbol{\mathcal{L}}'_1\boldsymbol{P}\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{U}}$. Merkitään vielä $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}_1'^{-1}\cdots\boldsymbol{\mathcal{L}}_{n-2}'^{-1}\boldsymbol{\mathcal{L}}_{n-1}^{-1}$, jolloin $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{U}} = \boldsymbol{P}\boldsymbol{\mathcal{M}}$. Operaattori $\boldsymbol{\mathcal{L}}$ koostuu Gaussin eliminaatiossa käytetyistä kertoimista

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{I}} + \boldsymbol{\eta}'_1\boldsymbol{\xi}_1^T + \boldsymbol{\eta}'_2\boldsymbol{\xi}_2^T + \cdots + \boldsymbol{\eta}'_{n-2}\boldsymbol{\xi}_{n-2}^T + \boldsymbol{\eta}_{n-1}\boldsymbol{\xi}_{n-1}^T.$$

2.3 Tuenta $\mathbb{R}$-lineaariselle LU-hajotelmalle

Seuraavaksi esitellään strategia, jonka avulla reaalilineaariselle operaattorille voidaan aina muodostaa LU-hajotelman kaltainen hajotelma. Ensiksi osoitetaan hajotelman olemassaolo, jonka jälkeen tarkastellaan kuinka sen voi laskea numeerisesti mielekkäällä tavalla. Tavoitteena on päästä eroon edellisessä kohdassa mainittujen kääntymättömien tukialkioiden ongelmasta.

Lemma 2.3.1. *Olkoot $\boldsymbol{\mu}_1 \neq \mathbf{0}$ ja $\boldsymbol{\mu}_2$ singulaarisia reaalilineaarisia skalaarioperaattoreita. Jos $\boldsymbol{\mu}_1 + q\boldsymbol{\mu}_2$ ei ole kääntyvä millään $q \in \mathbb{C}$, niin on olemassa $r \in \mathbb{C}$ siten, että $\boldsymbol{\mu}_2 - r\boldsymbol{\mu}_1 = \mathbf{0}$.*

Todistus. Jos $\boldsymbol{\mu}_2 = \mathbf{0}$, niin valitaan $r = 0$. Olkoon sitten $\boldsymbol{\mu}_2 \neq \mathbf{0}$. Merkitsemällä $\boldsymbol{\mu}_1(z) = u_1 z + v_1 \bar{z}$, $\boldsymbol{\mu}_2(z) = u_2 z + v_2 \bar{z}$ voidaan laskea

$$0 = |u_1 + qu_2|^2 - |v_1 + qv_2|^2 = (|u_1|^2 - |v_1|^2) + |q|^2(|u_2|^2 - |v_2|^2) + 2\,\mathrm{Re}\,(\bar{q}(u_1\bar{u}_2 - v_1\bar{v}_2)) = 2\,\mathrm{Re}\,(\bar{q}(u_1\bar{u}_2 - v_1\bar{v}_2)).$$

Valitsemalla $q = (u_1\bar{u}_2 - v_1\bar{v}_2)$ seuraa tästä $u_1\bar{u}_2 = v_1\bar{v}_2$. Olkoon $r = u_2/u_1$. Tällöin $u_2 - ru_1 = 0$ ja

$$v_2 - rv_1 = v_2 - \frac{u_2}{u_1}\frac{u_1\bar{u}_2}{\bar{v}_2} = v_2 - \frac{u_2\bar{u}_2 v_2}{\bar{v}_2 v_2} = 0.$$

Eli $\boldsymbol{\mu}_2 - r\boldsymbol{\mu}_1 = \mathbf{0}$. □

Lause 2.3.2. *Olkoon $\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^n$ reaalilineaarinen operaattori. Tällöin on olemassa alakolmio-operaattori $\boldsymbol{\mathcal{L}} : \mathbb{C}^n \to \mathbb{C}^n$, jonka lävistäjä koostuu skalaarisista yksikköoperaattoreista, yläkolmio-operaattori $\boldsymbol{\mathcal{U}} : \mathbb{C}^n \to \mathbb{C}^n$, yläkolmio-operaattori $\boldsymbol{\mathcal{V}} : \mathbb{C}^n \to \mathbb{C}^n$, jonka jokaisella rivillä on yksikkölävistäjäalkion lisäksi korkeintaan yksi toinen nollasta poikkeava kompleksiluku, ja permutaatio-operaattori $\boldsymbol{P} : \mathbb{C}^n \to \mathbb{C}^n$ siten, että*

$$\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{V}}\boldsymbol{P}\boldsymbol{\mathcal{M}}.$$

Todistus. Olkoon $\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^n$, $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mu}_{ij})$ reaalilineaarinen operaattori. Jos tämän ensimmäisessä sarakkeessa rivillä k on kääntyvä operaattori, niin kerrotaan $\boldsymbol{\mathcal{M}}$ vasemmalta rivinvaihto-operaattorilla $\boldsymbol{P}_1$, joka vaihtaa $\boldsymbol{\mathcal{M}}$:n ensimmäisen ja k:nnen rivin paikat keskenään. Tämän jälkeen voidaan suorittaa Gaussin eliminointi kuten edellä ja saadaan $\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$.

Jos ensimmäisessä sarakkeessa ei ole kääntyvää operaattoria, niin etsitään sieltä jokin nollasta poikkeava operaattori $\boldsymbol{\mu}_{k1}$. Jos tälläista ei ole, niin ensimmäinen sarake on jo eliminoitu ja voidaan siirtyä eliminoimaan seuraavaa saraketta, jolloin $\boldsymbol{\mathcal{L}}_1 = \boldsymbol{P}_1 = \boldsymbol{\mathcal{I}}$. Jos on ja löytyy jokin toinen $\boldsymbol{\mu}_{j1}$ ja $q \in \mathbb{C}$ siten, että $\boldsymbol{\mu}_{k1} + q\boldsymbol{\mu}_{j1}$ on kääntyvä operaattori, niin kerrotaan $\boldsymbol{\mathcal{M}}$ rivinvaihto-operaattorilla $\boldsymbol{P}_1$, joka vaihtaa ensimmäisen ja k:nnen rivin paikat, ja sen jälkeen kerrotaan operaattorilla $\boldsymbol{\mathcal{V}}_1 = \boldsymbol{\mathcal{I}} + \boldsymbol{\xi}_1\boldsymbol{\kappa}_1^T$, missä $\boldsymbol{\kappa}_j$ on $n \times 1$-operaattori, jonka j:s komponentti on q. Operaattorin $\boldsymbol{\mathcal{V}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$ vasemmassa yläkulmassa on nyt $\boldsymbol{\mu}_{k1} + q\boldsymbol{\mu}_{j1}$, joka on kääntyvä. Voidaan siis eliminoida Gaussin algoritmilla, jolloin saadaan $\boldsymbol{\mathcal{L}}_1\boldsymbol{\mathcal{V}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$.

Jos $\boldsymbol{\mu}_{k1} + q\boldsymbol{\mu}_{j1}$ ei ole kääntyvä millään $1 \leq j \leq n$, $j \neq k$, $q \in \mathbb{C}$, niin lemman 2.3.1 mukaan on olemassa luvut $r_j \in \mathbb{C}$ siten, että $\boldsymbol{\mu}_{j1} - r_j\boldsymbol{\mu}_{k1} = \mathbf{0}$. Olkoon $\boldsymbol{P}_1$ jälleen rivinvaihto-operaattori, joka vasemmalta kertomalla vaihtaa ensimmäisen ja k:nnen rivin paikat, ja olkoon $\boldsymbol{\mathcal{L}}_1 = \boldsymbol{\mathcal{I}} - \boldsymbol{\eta}_1\boldsymbol{\xi}_1^T$, jossa $\boldsymbol{\eta}_1$ sisältää komponentteina r_j:t paitsi ensimmäisessä komponentissa 0 ja k:nnessa r_1. Operaattorin $\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$ ensimmäisen sarakkeen, ensimmäisen rivin alapuoliset, komponentit voidaan nyt eliminoida kertomalla $\boldsymbol{\mathcal{L}}_1$:llä vasemmalta. Saadaan $\boldsymbol{\mathcal{L}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$.

Voidaan siirtyä eliminoimaan seuraava sarake samaan tapaan. Lopulta saadaan

$$\mathcal{L}_{n-1}\mathcal{V}_{n-1}P_{n-1}\mathcal{L}_{n-2}\mathcal{V}_{n-2}P_{n-2}\cdots\mathcal{L}_1\mathcal{V}_1P_1\mathcal{M}=\mathcal{U}.$$

Operaattorit $\mathcal{L}_k$ ja $\mathcal{V}_k$ ovat muotoa

$$\mathcal{L}_k=\mathcal{I}-\eta_k\xi_k^T, \qquad \mathcal{V}_k=\mathcal{I}+\xi_k\kappa_k^T \qquad k=1,\ldots,n-1,$$

missä η_k, κ_k ovat $n\times 1$-operaattoreita, joille $\xi_j^T\eta_k=\xi_j^T\kappa_k=0$, kun $j\leq k$.

Operaattorit $\mathcal{V}_k$ ja P_k voidaan kuljettaa operaattorien $\mathcal{L}_k$ perään seuraavasti. Pätee

$$\mathcal{V}_{n-1}P_{n-1}\mathcal{L}_{n-2}=\mathcal{V}_{n-1}\mathcal{L}'_{n-2}P_{n-1},$$

missä $\eta'_{n-2}=P_{n-1}\eta_{n-2}$ ja $\mathcal{L}'_{n-2}=\mathcal{I}-\eta'_{n-2}\xi_{n-2}^T$. Olkoon $\eta''_{n-2}=\mathcal{V}_{n-1}\eta'_{n-2}$ ja $\mathcal{L}''_{n-2}=\mathcal{I}-\eta''_{n-2}\xi_{n-2}^T$. Tällöin

$$\begin{aligned}\mathcal{V}_{n-1}\mathcal{L}'_{n-2}&=\mathcal{V}_{n-1}-\eta''_{n-2}\xi_{n-2}^T=\mathcal{I}+\xi_{n-1}^T\kappa_{n-1}-\eta''_{n-2}\xi_{n-2}^T\\&=(\mathcal{I}-\eta''_{n-2}\xi_{n-2}^T)(\mathcal{I}+\xi_{n-1}^T\kappa_{n-1})=\mathcal{L}''_{n-2}\mathcal{V}_{n-1}.\end{aligned}$$

Merkitään vielä $\eta''_k=\mathcal{V}_{n-1}P_{n-1}\mathcal{V}_{n-2}P_{n-2}\cdots\mathcal{V}_{k+1}P_{k+1}\eta_k$ ja $\mathcal{L}''_k=\mathcal{I}-\eta''_k\xi_k^T$. Tällöin edellä kuvatun kaltaisesti saadaan

$$\mathcal{L}_{n-1}\mathcal{L}''_{n-2}\cdots\mathcal{L}''_1\mathcal{V}_{n-1}P_{n-1}\mathcal{V}_{n-2}P_{n-2}\cdots\mathcal{V}_1P_1\mathcal{M}=\mathcal{U}.$$

Operaattorit P_k voidaan vielä kuljettaa operaattoreiden $\mathcal{V}_k$ eteen. Kun merkitään $\kappa'_{n-2}=P_{n-1}\kappa_{n-2}$ ja $\mathcal{V}'_{n-2}=\mathcal{I}+\xi_{n-2}\kappa'^T_{n-2}$, niin

$$\begin{aligned}P_{n-1}\mathcal{V}_{n-2}&=P_{n-1}+P_{n-1}\xi_{n-2}\kappa_{n-2}^T=P_{n-1}+\xi_{n-2}\kappa_{n-2}^T\\&=P_{n-1}+\xi_{n-2}(P_{n-1}^2\kappa_{n-2})^T=P_{n-1}+\xi_{n-2}(P_{n-1}\kappa_{n-2})^TP_{n-1}^T\\&=P_{n-1}+\xi_{n-2}\kappa'^T_{n-2}P_{n-1}=\mathcal{V}'_{n-2}P_{n-1}.\end{aligned}$$

Jatketaan tätä kunnes kaikki P_k:t ovat perässä. Kun $\kappa'_k=P_{n-1}P_{n-2}\cdots P_{k+1}\kappa_k$ ja $\mathcal{V}'_k=\mathcal{I}+\xi_k\kappa'^T_k$, niin

$$\mathcal{L}_{n-1}\mathcal{L}''_{n-2}\cdots\mathcal{L}''_1\mathcal{V}_{n-1}\mathcal{V}'_{n-2}\cdots\mathcal{V}'_1P_{n-1}P_{n-2}\cdots P_1\mathcal{M}=\mathcal{U}.$$

Merkitsemällä

$$\mathcal{L}=\mathcal{L}_1''^{-1}\cdots\mathcal{L}''^{-1}_{n-2}\mathcal{L}^{-1}_{n-1},\ \mathcal{V}=\mathcal{V}_{n-1}\mathcal{V}'_{n-2}\cdots\mathcal{V}'_1 \text{ ja } P=P_{n-1}P_{n-2}\cdots P_1$$

voidaan edellä oleva lausua $\mathcal{L}\mathcal{U}=\mathcal{V}P\mathcal{M}$. Tässä operaattori $\mathcal{L}$ voidaan muodostaa suoraan Gaussin eliminoinnissa käytetyistä kertoimista kuten aiemmin:

$$\mathcal{L}=\mathcal{I}+\eta''_1\xi_1^T+\cdots+\eta''_{n-2}\xi_{n-2}^T+\eta''_{n-1}\xi_{n-1}^T.$$

Myös operaattori $\mathcal{V}$ voidaan muodostaa suoraan seuraavasti. Merkitään $\mathcal{S}_k=\mathcal{V}'_k\cdots\mathcal{V}'_2\mathcal{V}'_1$, jolloin induktiolla voidaan osoittaa

$$\mathcal{S}_k=\mathcal{I}+\xi_1\kappa'^T_1+\xi_2\kappa'^T_2+\cdots+\xi_k\kappa'^T_k.$$

Nimittäin selvästi $\mathcal{S}_1=\mathcal{V}'_1$ on tätä muotoa. Jos $\mathcal{S}_k$ on yllä olevaa muotoa, niin

$$\begin{aligned}\mathcal{S}_{k+1}=\mathcal{V}'_{k+1}\mathcal{S}_k&=(\mathcal{I}+\xi_{k+1}\kappa'^T_{k+1})\mathcal{S}_k=\mathcal{I}+\xi_1\kappa'^T_1+\xi_2\kappa'^T_2+\cdots+\xi_k\kappa'^T_k+\\&\xi_{k+1}\kappa'^T_{k+1}+\xi_{k+1}(\kappa'^T_{k+1}\xi_1)\kappa'^T_1+{+}\xi_{k+1}(\kappa'^T_{k+1}\xi_2)\kappa'^T_2+\cdots+\xi_{k+1}(\kappa'^T_{k+1}\xi_k)\kappa'^T_k\\&=\mathcal{I}+\xi_1\kappa'^T_1+\xi_2\kappa'^T_2+\cdots+\xi_k\kappa'^T_k+\xi_{k+1}\kappa'^T_{k+1},\end{aligned}$$

joka on haluttua muotoa. Täten

$$\mathcal{V}=\mathcal{I}+\xi_1\kappa'^T_1+\xi_2\kappa'^T_2+\cdots+\xi_{n-2}\kappa'^T_{n-2}+\xi_{n-1}\kappa_{n-1}^T.$$

Lause on nyt osoitettu. □

2.3.1 $\mathbb{R}$-lineaarisesti tuetun LU-hajotelman laskeminen

Esitellään sitten eräs tapa laskea edellä kuvattu hajotelma reaalilineaarisille operaattoreille tavalla, joka pyrkii ottamaan huomioon numeeriset näkökohdat. Oletetaan seuraavassa, että operaattori $\boldsymbol{\mathcal{M}}$ on kääntyvä.

Lauseen 2.3.2 mukaiset $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{V}}$ ja $\boldsymbol{P}$ ovat kääntyviä, joten myös $\boldsymbol{\mathcal{U}}$ on kääntyvä. Erityisesti tästä seuraa, että kaikki $\boldsymbol{\mathcal{U}}$:n skalaariset lävistäjäoperaattorit ovat kääntyviä. Täten lauseen 2.3.2 todistuksen alussa sarakkeelta löytyy joko heti kääntyvä operaattori tai sitten nollasta poikkeava $\boldsymbol{\mu}_{k1}$ ja $q \in \mathbb{C}$, $j \neq k$ siten, että $\boldsymbol{\mu}_{k1} + q\boldsymbol{\mu}_{j1}$ on kääntyvä.

Skalaarioperaattorille $\boldsymbol{\mu}(z) = uz + v\bar{z}$ määritellään kohdan 2.2 tavoin suure $N(\boldsymbol{\mu}) = |u|^2 - |v|^2$ ja käytetään seuraavaa strategiaa. Etsitään ensin operaattorin $\boldsymbol{\mathcal{M}}$ ensimmäiseltä sarakkeelta alkio $\boldsymbol{\mu}_{k1}$ siten, että $|N(\boldsymbol{\mu}_{k1})|$ on suurin mahdollinen. Seuraavaksi lasketaan jokaiselle $i, j \in \mathbb{N}$, $1 \leq i, j \leq n$, $i \neq j$ luku $N(\boldsymbol{\mu}_{i1} + q_{ij}\boldsymbol{\mu}_{j1})$, jossa $q_{ij} = s_{ij}r_{ij}$, $r_{ij} = (u_{i1}\bar{u}_{j1} - v_{i1}\bar{v}_{j1})$ ja $s_{ij} \in \mathbb{R}$ vielä määräämätön reaaliluku. Tällöin

$$\begin{aligned} N(\boldsymbol{\mu}_{i1} + q_{ij}\boldsymbol{\mu}_{j1}) &= |u_{i1} + q_{ij}u_{j1}|^2 - |v_{i1} + q_{ij}v_{j1}|^2 \\ &= (|u_{i1}|^2 - |v_{i1}|^2) + |q_{ij}|^2(|u_{j1}|^2 - |v_{j1}|^2) + \\ &\quad 2\,\mathrm{Re}\,(\bar{q}_{ij}(u_{i1}\bar{u}_{j1} - v_{i1}\bar{v}_{j1})) = N(\boldsymbol{\mu}_{i1}) + s_{ij}^2|r_{ij}|^2 N(\boldsymbol{\mu}_{j1}) + 2s_{ij}|r_{ij}|^2. \end{aligned}$$

Huomaa, että pidettäessä $|q_{ij}|$ kiinnitettynä, tehty valinta q_{ij}:lle maksimoi arvon $N(\boldsymbol{\mu}_{i1} + q_{ij}\boldsymbol{\mu}_{j1})$.

Valittaessa lukuja s_{ij} pyritään siihen ettei operaattorin $\boldsymbol{\mathcal{V}}_1\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$ ensimmäisen rivin alkioiden suuruusluokka kasvaisi kohtuuttomasti operaattorin $\boldsymbol{P}_1\boldsymbol{\mathcal{M}}$ ensimmäiseen riviin nähden. Valitaan tässä yksinkertaisesti joko $s_{ij} = 1$ tai $s_{ij} = -1$ sen perusteella kummalla valinnalla lausekkeen $|N(\boldsymbol{\mu}_{i1} + q_{ij}\boldsymbol{\mu}_{j1})|$ arvo on suurempi.

Lopuksi poimitaan suurempi arvoista $|N(\boldsymbol{\mu}_{k1})|$ ja $|N(\boldsymbol{\mu}_{i1} + q_{ij}\boldsymbol{\mu}_{j1})|$. Liitteessä B oleva `MATLAB`-funktio `rl_luvp` laskee yllä kuvatulla tavalla hajotelman $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{V}}\boldsymbol{P}\boldsymbol{\mathcal{M}}$.

Luku 3

Iteratiivinen ratkaiseminen

3.1 Johdanto

Olkoon $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, $\boldsymbol{b} \in \mathbb{C}^n$ ja tehtävänä ratkaista lineaarinen yhtälöryhmä

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{3.1}$$

Tätä tarkoitusta varten edellisessä luvussa esiteltiin Gaussin menetelmään perustuva $\boldsymbol{A}$:n LU-hajotelma. Tämä algoritmi vaatii $O(n^3)$ laskuoperaatiota ja on esimerkki ns. suorasta menetelmästä lineaarisen yhtälöryhmän ratkaisemiseksi. Muistia se vaatii $O(n^2)$ alkiota. Suurilla n toinen tai molemmat näistä vaatimuksista saattavat ylittää käytettävissä olevat tietokoneresurssit. Erityisesti osittaisdifferentiaaliyhtälöiden ja integraaliyhtälöiden diskretoinnista seuraavissa yhtälöryhmissä n kasvaa haettaessa lisää tarkkuutta lopulta resurssit ylittäen. Toisaalta näiden diskretointien matriiseilla $\boldsymbol{A}$ on usein rakenne, jota voidaan käyttää hyödyksi. Matriisia $\boldsymbol{A}$ kutsutaan harvaksi, jos niin suuri määrä sen alkioista on nollia, että tämän hyödyntäminen on mahdollista.

Iteratiivisissa menetelmissä lähdetään liikkeelle ratkaisun approksimaatiosta, jota pyritään tarkentamaan algoritmin jokaisella iteraatiokierroksella. Iterointi pysäytetään, kun riittävä tarkkuus on saavutettu. Erilaisia menetelmiä on 1950-luvulta lähtien esitetty lukuisia ja niiden suppenemisnopeudet kohti ratkaisua riippuvat paitsi menetelmästä myös matriisin $\boldsymbol{A}$ ominaisuuksista. Suppenemisnopeuteen voidaan huomattavasti vaikuttaa myös ns. pohjustuksella, mistä enemmän seuraavassa luvussa.

Usein käytännön tehtävissä esiintyvien yhtälöryhmien matriiseilla $\boldsymbol{A}$ on ominaisuuksia, jotka tekevät iteratiiviset menetelmät suoria menetelmiä laskennallisesti edullisemmiksi. Tyypillisiä iteratiivisten menetelmien perusoperaatioita jokaisella iteraatiokierroksella ovat matriisilla $\boldsymbol{A}$ kertominen ja sisätulon laskeminen. Harvoista matriiseista nauhamatriisien tulot vektoreiden kanssa vievät vain $O(n)$ laskuoperaatiota. Suorat menetelmät eivät myöskään tule kyseeseen mikäli itse matriisia $\boldsymbol{A}$ ei ole mielekästä muodostaa ja tallentaa tietokoneen muistiin, mutta operaatio $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$ voidaan suorittaa mille tahansa vektorille $\boldsymbol{x}$. Toisaalta joskus yhtälön ratkaisun ei tarvitse olla tarkka, jolloin iteroinnin voi pysäyttää ratkaisun saavutettua saman virherajan kuin mittausarvoista peräisin olevilla $\boldsymbol{A}$:lla ja $\boldsymbol{b}$:llä. Suoralla menetelmäl-

lä ratkaiseminen täytyy viedä loppuun asti, koska laskennan välitulokset eivät ole käyttökelpoisia likimääräisratkaisuja.

Myös tässä luvussa esitellään aluksi joitakin iteratiivisia menetelmiä matriiseille ja sen jälkeen siirrytään käsittelemään reaalilineaaristen operaattoreiden menetelmiä.

3.2 Petrov-Galerkinin menetelmä

Abstrakti Galerkinin menetelmä on pohjana useille eri menetelmille. Yhtälö (3.1) voidaan kirjoittaa yhtäpitävästi muotoon

$$\langle \boldsymbol{Ax}, \boldsymbol{y} \rangle = \langle \boldsymbol{b}, \boldsymbol{y} \rangle \qquad \text{kaikilla } \boldsymbol{y} \in \mathbb{C}^n.$$

Notaatio $\langle \cdot, \cdot \rangle$ tarkoittaa $\mathbb{C}^n$:n sisätuloa $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^* \boldsymbol{x}$. Kirjoitetaan edellinen yhtälö muotoon

$$\langle \boldsymbol{b} - \boldsymbol{Ax}, \boldsymbol{y} \rangle = 0 \qquad \text{kaikilla } \boldsymbol{y} \in \mathbb{C}^n. \tag{3.2}$$

Korvaamalla tässä avaruus $\mathbb{C}^n$ sopivasti aliavaruuksilla, saadaan approksimoiva yhtälö. Olkoot X_k, $Y_k \subset \mathbb{C}^n$ joitakin $\mathbb{C}^n$:n aliavaruuksia ja $\boldsymbol{x}_0$ ratkaisun alkuarvaus. Avaruuksien X_k, Y_k dimensiot ovat tyypillisesti huomattavasti pienemmät kuin n. Petrov-Galerkinin ehto ratkaisun approksimaatioksi $\boldsymbol{x}_k \in \boldsymbol{x}_0 + X_k$ on

$$\langle \boldsymbol{b} - \boldsymbol{Ax}_k, \boldsymbol{y} \rangle = 0 \qquad \text{kaikilla } \boldsymbol{y} \in Y_k. \tag{3.3}$$

Ratkaisua $\boldsymbol{x}_k$ etsitään siis affiinista aliavaruudesta $\boldsymbol{x}_0 + X_k$ siten, että jäännös $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{Ax}_k$ on kohtisuorassa Y_k:ta vastaan.

Ratkaisun $\boldsymbol{x}_k$ laskemiseksi, olkoot matriisin $\boldsymbol{V}_k \in \mathbb{C}^{n \times k}$ sarakkeet avaruuden X_k jotkin kantavektorit ja matriisin $\boldsymbol{W}_k \in \mathbb{C}^{n \times k}$ sarakkeet Y_k:n kanta. Olkoon $\boldsymbol{z}_k \in \mathbb{C}^k$ siten, että

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k. \tag{3.4}$$

Merkitään vielä $\boldsymbol{v} = \boldsymbol{b} - \boldsymbol{Ax}_0$, jolloin yhtälö (3.3) saadaan muotoon

$$\boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k \boldsymbol{z}_k = \boldsymbol{W}_k^* \boldsymbol{v}. \tag{3.5}$$

Ratkaisemalla tästä $\boldsymbol{z}_k$ ja sijoittamalla kaavaan (3.4) saadaan yhtälön (3.1) ratkaisun approksimaatio $\boldsymbol{x}_k$.

Matriisi $\boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k$ ei välttämättä ole kääntyvä vaikka $\boldsymbol{A}$ olisikin. Seuraavan lauseen kahdessa tapauksessa näin kuitenkin on.

Lause 3.2.1. *Olkoon $\boldsymbol{V} \in \mathbb{C}^{n \times k}$ aliavaruuden X_k kanta ja $\boldsymbol{W} \in \mathbb{C}^{n \times k}$ aliavaruuden Y_k kanta. Tällöin $\boldsymbol{W}^* \boldsymbol{AV}$ on kääntyvä, jos*

(1) $\langle \boldsymbol{Ax}, \boldsymbol{x} \rangle \neq 0$ kaikilla $\boldsymbol{x} \neq 0$ ja $X_k = Y_k$, tai

(2) $\boldsymbol{A}$ on kääntyvä ja $Y_k = \boldsymbol{A} X_k$.

Todistus. Ensimmäisessä tapauksessa on olemassa kannanvaihtomatriisi $\boldsymbol{K} \in \mathbb{C}^{k \times k}$ siten, että $\boldsymbol{W} = \boldsymbol{VK}$. Tällöin $\boldsymbol{W}^* \boldsymbol{AV} = \boldsymbol{K}^* \boldsymbol{V}^* \boldsymbol{AV}$. Koska

$$\langle \boldsymbol{V}^* \boldsymbol{AV} \boldsymbol{z}, \boldsymbol{z} \rangle = \langle \boldsymbol{AV} \boldsymbol{z}, \boldsymbol{V} \boldsymbol{z} \rangle \neq 0 \qquad \text{kaikilla } \boldsymbol{z} \in \mathbb{C}^k,\ \boldsymbol{z} \neq \boldsymbol{0},$$

niin matriisi $\boldsymbol{V}^*\boldsymbol{A}\boldsymbol{V}$ on kääntyvä, koska sen nolla-avaruus sisältää vain nollavektorin. Koska myös kannanvaihtomatriisi $\boldsymbol{K}$ on kääntyvä, väite on saatu osoitettua.

Toista tapausta varten huomataan, että $\boldsymbol{A}\boldsymbol{V}$ on Y_k:n kanta. Täten on olemassa kannanvaihtomatriisi $\boldsymbol{K} \in \mathbb{C}^{k\times k}$ siten, että $\boldsymbol{A}\boldsymbol{V} = \boldsymbol{W}\boldsymbol{K}$. Tällöin $\boldsymbol{W}^*\boldsymbol{A}\boldsymbol{V} = \boldsymbol{W}^*\boldsymbol{W}\boldsymbol{K}$. Matriisi $\boldsymbol{K}$ on kääntyvä ja myös $\boldsymbol{W}^*\boldsymbol{W}$ on kääntyvä, koska

$$\langle \boldsymbol{W}^*\boldsymbol{W}\boldsymbol{z}, \boldsymbol{z}\rangle = \langle \boldsymbol{W}\boldsymbol{z}, \boldsymbol{W}\boldsymbol{z}\rangle = \|\boldsymbol{W}\boldsymbol{z}\|^2 > 0 \qquad \text{kaikilla } \boldsymbol{z} \in \mathbb{C}^k,\ \boldsymbol{z} \neq \boldsymbol{0}.$$

□

Petrov-Galerkinin yhtälön ratkaisulle pätee seuraavat yleiset optimaalisuustulokset.

Lause 3.2.2. *Olkoon $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ hermiittinen ja positiividefiniitti matriisi ja $X_k = Y_k$. Tällöin Petrov-Galerkinin yhtälön ratkaisun $\boldsymbol{x}_k$ virhe on kohtisuorassa aliavaruutta Y_k vastaan $\boldsymbol{A}$-normissa eli*

$$\langle \boldsymbol{x}_* - \boldsymbol{x}_k, \boldsymbol{y}\rangle_{\boldsymbol{A}} = 0 \qquad \textit{kaikilla } \boldsymbol{y} \in Y_k,$$

missä $\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\boldsymbol{A}} = \langle \boldsymbol{A}\boldsymbol{u}, \boldsymbol{v}\rangle$ ja $\boldsymbol{x}_*$ on yhtälön (3.1) tarkka ratkaisu. Approksimaatio $\boldsymbol{x}_k$ minimoi virheen $\boldsymbol{A}$-normissa

$$\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_{\boldsymbol{A}} = \min_{\boldsymbol{y}\in X_k} \|\boldsymbol{x}_* - \boldsymbol{y}\|_{\boldsymbol{A}}.$$

Todistus. Asetetaan yhtälössä (3.2) $\boldsymbol{x} = \boldsymbol{x}_*$ ja vähennetään se puolittain (3.3):sta, jolloin

$$\langle \boldsymbol{A}(\boldsymbol{x}_* - \boldsymbol{x}_k), \boldsymbol{y}\rangle = 0 \qquad \text{kaikilla } \boldsymbol{y} \in Y_k.$$

Virheen minimoituminen seuraa Pythagoraan lauseesta.

$$\|\boldsymbol{x}_* - \boldsymbol{y}\|_{\boldsymbol{A}}^2 = \|\boldsymbol{x}_* - \boldsymbol{x}_k + \boldsymbol{x}_k - \boldsymbol{y}\|_{\boldsymbol{A}}^2 = \|\boldsymbol{x}_* - \boldsymbol{x}_k\|_{\boldsymbol{A}}^2 + \|\boldsymbol{x}_k - \boldsymbol{y}\|_{\boldsymbol{A}}^2 \qquad (\boldsymbol{y} \in X_k),$$

missä on käytetty kohtisuoruutta $(\boldsymbol{x}_* - \boldsymbol{x}_k) \perp (\boldsymbol{x}_k - \boldsymbol{y})$. Virhe minimoituu, kun $\boldsymbol{y} = \boldsymbol{x}_k$. □

Toisessa tapauksessa ($Y_k = \boldsymbol{A}X_k$) nähdään yhtälöstä (3.3), että approksimaation $\boldsymbol{x}_k$ antama jäännös $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ on kohtisuorassa avaruutta $\boldsymbol{A}X_k$ vastaan. Pythagoraan lauseesta seuraa (kuten edellisen lauseen todistuksessa), että jäännöksen 2-normi $\|\boldsymbol{r}_k\|_2$ on minimissään täsmälleen silloin, kun $\boldsymbol{r}_k \perp \boldsymbol{A}X_k$. Saadaan seuraava lause.

Lause 3.2.3. *Olkoon $Y_k = \boldsymbol{A}X_k$. Tällöin Petrov-Galerkinin yhtälön ratkaisun $\boldsymbol{x}_k$ jäännös $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ on kohtisuorassa avaruutta $\boldsymbol{A}\boldsymbol{x}_k$ vastaan. Tämä tapahtuu täsmälleen silloin, kun jäännöksen 2-normi on minimissään*

$$\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\|_2 = \min_{\boldsymbol{y}\in \boldsymbol{A}X_k} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}\|_2.$$

Seuraavaksi esitellään Krylovin aliavaruudet, joita käyttäen yhdessä Petrov-Galerkinin yhtälön kanssa saadaan muodostettua iteratiivisia ratkaisumenetelmiä yhtälölle (3.1). Liittogradienttimenetelmä on esimerkki tällaisesta menetelmästä ($Y_k = X_k$ ja $\boldsymbol{A}$ hermiittinen ja positiividefiniitti) ja sille pätee lause 3.2.2. GMRES on esimerkki menetelmästä, jolle pätee lause 3.2.3. Näistä esitellään vain GMRES, minkä jälkeen siirrytään käsittelemään reaalilineaarisia menetelmiä.

3.3 Matriisien Krylovin aliavaruudet

Valitaan X_k:ksi Krylovin aliavaruudet

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \operatorname{span}\left\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \boldsymbol{A}^2\boldsymbol{v}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{v}\right\}, \qquad k = 1, 2, \ldots$$

Näille pätee $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) \subset \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v})$ kaikilla k. Vektorin $\boldsymbol{v}$ minimipolynomi $\boldsymbol{A}$:n suhteen on pienintä astetta oleva mooninen polynomi p (korkeimman asteen termin kerroin on yksi) siten, että $p(\boldsymbol{A})\boldsymbol{v} = 0$. Tällainen minimipolynomi on aina olemassa, koska erityisesti $\boldsymbol{A}$:n karakteristisella polynomilla p on $p(\boldsymbol{A}) = 0$ Cayley-Hamiltonin lauseen mukaan. Seuraava lause kertoo, että Krylovin aliavaruudet kasvavat aidosti minimipolynomin astelukuun asti eivätkä sen jälkeen muutu.

Lause 3.3.1. *Olkoot $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{v} \in \mathbb{C}^n$ ja p on $\boldsymbol{v}$:n minimipolynomi. Kun merkitään $\mu = \deg p$, niin*

(1) $\dim \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = k$, *kun* $k \leq \mu$,

(2) $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_\mu(\boldsymbol{A}, \boldsymbol{v})$, *kun* $k \geq \mu$.

Todistus. Olkoon $k \leq \mu$. Selvästi $\dim \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) \leq k$. Kun jokin kertoimista $c_0, c_1, \ldots, c_{k-1}$ poikkeaa nollasta, täytyy olla $c_0\boldsymbol{v} + c_1\boldsymbol{A}\boldsymbol{v} + \cdots + c_{k-1}\boldsymbol{A}^{k-1}\boldsymbol{v} \neq \boldsymbol{0}$. Muutoin tästä voitaisiin muodostaa minimipolynomi, jonka asteluku olisi pienempi kuin μ vastoin p:n valintaa. Vektorit $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{v}$ ovat siten lineaarisesti riippumattomat, joten kohta (1) on osoitettu.

Olkoon sitten $k \geq \mu$. Minimipolynomin mukaan $\boldsymbol{A}^\mu\boldsymbol{v}$ on lineaarikombinaatio vektoreista $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \ldots, \boldsymbol{A}^{\mu-1}\boldsymbol{v}$. Huomaamalla, että $\boldsymbol{A}^j\boldsymbol{v} = \boldsymbol{A}\boldsymbol{A}^{j-1}\boldsymbol{v}$, saadaan induktiolla, että myös $\boldsymbol{A}^k\boldsymbol{v}$ on lineaarikombinaatio samoista vektoreista. Kohta (2) on täten osoitettu. □

Krylovin aliavaruuksiin $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ voidaan muodostaa ortonormaali kanta Arnoldin menetelmällä. Olkoon ensimmäinen kantavektori $\boldsymbol{q}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|$. Oletetaan sitten, että kantavektorit $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_j$ on laskettu ja näytetään kuinka $\boldsymbol{q}_{j+1}$ saadaan. Lasketaan vektori $\boldsymbol{A}\boldsymbol{q}_j$ ja ortogonalisoidaan se käyttäen Gram-Schmidt menetelmää

$$h_{ij} = \langle \boldsymbol{A}\boldsymbol{q}_j, \boldsymbol{q}_i \rangle \qquad (i = 1, 2, \ldots, j) \tag{3.6}$$

$$\boldsymbol{w}_j = \boldsymbol{A}\boldsymbol{q}_j - \sum_{i=1}^{j} h_{ij}\boldsymbol{q}_i \tag{3.7}$$

$$h_{j+1,j} = \|\boldsymbol{w}_j\| \tag{3.8}$$

Menetelmä pysäytetään, jos $\boldsymbol{w}_j = 0$. Tällöin sanotaan menetelmän murtuneen ja tätä j:nnettä askelta murtumisaskeleeksi. Muutoin asetetaan $\boldsymbol{q}_{j+1} = \boldsymbol{w}_j/\|\boldsymbol{w}_j\|$.

Algoritmi 3.3.1 laskee saman kannan käyttäen muunnettua Gram-Schmidt menetelmää. Tämä on tarkassa aritmetiikassa yhtäpitävä edellä kuvatun kanssa, mutta on numeerisessa laskennassa tapahtuvien pyöristysvirheiden läsnäollessa luotettavampi. Vektorin $\boldsymbol{A}\boldsymbol{q}_j$ ortogonalisointi tapahtuu siinä yksitellen, jolloin vektoreissa $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, $\ldots$, $\boldsymbol{q}_j$ jo oleva pyöristysvirheistä aiheutunut epäortogonaalisuus tulee huomioiduksi.

Algoritmi 3.3.1 Arnoldin menetelmä ortonormaalin kannan muodostamiseksi Krylovin aliavaruudelle $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ käyttäen muunnettua Gram-Schmidt menetelmää. Laskee ortonormaalit kantavektorit $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots$ ja tallentaa ortogonalisoinnissa käytetyt sisätulot matriisiin $\boldsymbol{H} = (h_{ij})$. Algoritmi pysähtyy muodostettuaan $\min\{k+1, \dim \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})\}$ kantavektoria.

1: $\boldsymbol{q}_1 \leftarrow \boldsymbol{v}/\|\boldsymbol{v}\|$
2: **for** $j = 1$ to k **do**
3: $\quad \boldsymbol{w} \leftarrow \boldsymbol{A}\boldsymbol{q}_j$
4: $\quad$ **for** $i = 1$ to j **do**
5: $\quad\quad h_{ij} \leftarrow \langle \boldsymbol{w}, \boldsymbol{q}_i \rangle$
6: $\quad\quad \boldsymbol{w} \leftarrow \boldsymbol{w} - h_{ij}\boldsymbol{q}_i$
7: $\quad$ **end for**
8: $\quad h_{j+1,j} \leftarrow \|\boldsymbol{w}\|$
9: $\quad$ **if** $h_{j+1,j} = 0$ **then**
10: $\quad\quad$ lopeta algoritmi
11: $\quad$ **end if**
12: $\quad \boldsymbol{q}_{j+1} \leftarrow \boldsymbol{w}/h_{j+1,j}$
13: **end for**

Lause 3.3.2. *Oletetaan, että Arnoldin menetelmällä on saatu muodostettua vektorit $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$. Nämä vektorit muodostavat ortonormaalin kannan avaruuteen $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$.*

Todistus. Vektorit ovat ortonormaalit Gram-Schmidt prosessin ja normeerauksen myötä. Osoitetaan, että jokaiselle $\boldsymbol{q}_j$ on olemassa astetta $j-1$ oleva polynomi p_{j-1} siten, että $\boldsymbol{q}_j = p_{j-1}(\boldsymbol{A})\boldsymbol{v}$. Tämä on selvästi totta $\boldsymbol{q}_1$:lle. Oletetaan sitten, että näin on $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_j$:lle ja osoitetaan todeksi myös $\boldsymbol{q}_{j+1}$:lle. Kaavasta (3.7) saadaan

$$\boldsymbol{q}_{j+1} = \tfrac{1}{\|\boldsymbol{w}_j\|}\boldsymbol{A}\boldsymbol{q}_j - \sum_{i=1}^{j} \tfrac{1}{\|\boldsymbol{w}_j\|} h_{ij}\boldsymbol{q}_i = \tfrac{1}{\|\boldsymbol{w}_j\|}\boldsymbol{A}p_{j-1}(\boldsymbol{A})\boldsymbol{v} - \sum_{i=1}^{j} \tfrac{1}{\|\boldsymbol{w}_j\|} h_{ij}p_{i-1}(\boldsymbol{A})\boldsymbol{v},$$

mistä nähdään, että oikealla puolella on j-asteinen polynomi.

Ollaan saatu, että $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$ kuuluvat avaruuteen $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$. Krylovin aliavaruuden määritelmästä saadaan $\dim \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) \leq k$, joten vektorit virittävät avaruuden ja $\dim \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = k$. Lisäksi nähdään, että $\boldsymbol{v}$:n minimipolynomi on vähintään astetta k. □

Lause 3.3.3. *Arnoldin menetelmässä $\boldsymbol{w}_j = 0$ jos ja vain jos j on $\boldsymbol{v}$:n minimipolynomin asteluku.*

Todistus. Oletetaan aluksi $\boldsymbol{w}_j = 0$. Kuten lauseen 3.3.2 todistuksessa, olkoon p_{j-1} astetta j oleva polynomi siten, että $\boldsymbol{q}_j = p_{j-1}(\boldsymbol{A})\boldsymbol{v}$. Kaavasta (3.7) saadaan

$$0 = \boldsymbol{w}_j = \boldsymbol{A}\boldsymbol{q}_j - \sum_{i=1}^{j} h_{ij}\boldsymbol{q}_i = \boldsymbol{A}p_{j-1}(\boldsymbol{A})\boldsymbol{v} - \sum_{i=1}^{j} h_{ij}p_{i-1}(\boldsymbol{A})\boldsymbol{v},$$

jonka oikealla puolella on j-asteinen polynomi. Siispä $\boldsymbol{v}$:n minimipolynomin asteluku on korkeintaan j. Koska $\boldsymbol{w}_{j-1} \neq 0$, on toisaalta saatu muodostettua $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_j$, joten

lauseen 3.3.2 perusteella $\dim \mathcal{K}_j(\boldsymbol{A}, \boldsymbol{v}) = j$ ja lauseen 3.3.1 mukaan minimipolynomin asteluku ei voi olla pienempi kuin j.

Oletetaan sitten, että j on $\boldsymbol{v}$:n minimipolynomin asteluku. Lauseiden 3.3.1 ja 3.3.2 perusteella Arnoldin menetelmällä on mahdotonta muodostaa enemmän kuin j ortonormaalia vektoria. Täytyy siis olla $\boldsymbol{w}_i = 0$ jollakin $i \leq j$. Toisaalta, jos $i < j$, niin tämän todistuksen ensimmäisen osan mukaan asteluvunkin täytyisi olla i. Täten $\boldsymbol{w}_j = 0$. □

Edeltävien lauseiden sisältönä on, että Arnoldin menetelmä toimii niin pitkälle kuin mahdollista. Se pysähtyy täsmälleen silloin, kun Krylovin aliavaruuksien jono ($k = 1, 2, \dots$) lakkaa kasvamasta.

Olkoot $\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_k$ muodostettu Arnoldin menetelmällä ja asetetaan ne sarakkeiksi matriisin $\boldsymbol{Q}_k = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_k \end{bmatrix}$. Olkoon $\widehat{\boldsymbol{H}}_k$ Arnoldin menetelmän alkioista h_{ij} muodostettu $(k+1) \times k$-matriisi, $(\widehat{\boldsymbol{H}}_k)_{ij} = h_{ij}$ $(i \leq j+1)$ ja muut alkiot nollia. Olkoon vielä $\boldsymbol{H}_k$ se $k \times k$-matriisi, kun matriisista $\widehat{\boldsymbol{H}}_k$ poistetaan viimeinen rivi. Kaavasta (3.7) saadaan

$$h_{j+1,j}\boldsymbol{q}_{j+1} = \boldsymbol{A}\boldsymbol{q}_j - \sum_{i=1}^{j} h_{ij}\boldsymbol{q}_i \qquad \Leftrightarrow \qquad \boldsymbol{A}\boldsymbol{q}_j = \sum_{i=1}^{j+1} h_{ij}\boldsymbol{q}_i,$$

joka voidaan kirjoittaa matriisimuotoihin

$$\boldsymbol{A}\boldsymbol{Q}_j = \boldsymbol{Q}_{j+1}\widehat{\boldsymbol{H}}_j, \tag{3.9}$$

$$\boldsymbol{A}\boldsymbol{Q}_j = \boldsymbol{Q}_j\boldsymbol{H}_j + \boldsymbol{w}_j\boldsymbol{e}_j^T, \tag{3.10}$$

missä $\boldsymbol{e}_j$ on $\mathbb{C}^j$:n j:s luonnollinen kantavektori. Kertomalla jälkimmäinen puolittain $\boldsymbol{Q}_j^*$:llä ja hyödyntämällä $\boldsymbol{Q}_j$:n sarakkeiden ortonormaalisuutta, saadaan lisäksi

$$\boldsymbol{Q}_j^*\boldsymbol{A}\boldsymbol{Q}_j = \boldsymbol{H}_j. \tag{3.11}$$

3.4 GMRES-menetelmä

GMRES (Generalized Minimal Residual) menetelmässä likimääräisratkaisua $\boldsymbol{x}_k$ etsitään avaruudesta $\boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ ja vaaditaan, että jäännösvektorin $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ 2-normi $\|\boldsymbol{r}_k\|$ on pienin mahdollinen. Tässä $\boldsymbol{v} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ ja $\boldsymbol{x}_0$ on alkuarvaus. Lauseen 3.2.3 mukaan normin $\|\boldsymbol{r}_k\|$ minimointi on yhtäpitävä kohtisuoruuden ehdon (3.3) kanssa, kun merkitään $X_k = \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$, $Y_k = \boldsymbol{A}X_k$.

Valitaan k ja lasketaan Arnoldin menetelmällä - käyttäen esimerkiksi algoritmia 3.3.1 - ortonormaalit kantavektorit Krylovin aliavaruudelle $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ ja asetetaan ne matriisin $\boldsymbol{Q}_k$ sarakkeiksi. Tällöin yhtälöstä (3.9)

$$\begin{aligned} \|\boldsymbol{r}_k\| &= \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\| = \|\boldsymbol{v} - \boldsymbol{A}\boldsymbol{Q}_k\boldsymbol{z}_k\| = \|\boldsymbol{v} - \boldsymbol{Q}_{k+1}\widehat{\boldsymbol{H}}_k\boldsymbol{z}_k\| \\ &= \left\|\boldsymbol{Q}_{k+1}(\|\boldsymbol{v}\|\boldsymbol{e}_1 - \widehat{\boldsymbol{H}}_k\boldsymbol{z}_k)\right\| = \left\|\|\boldsymbol{v}\|\boldsymbol{e}_1 - \widehat{\boldsymbol{H}}_k\boldsymbol{z}_k\right\|, \end{aligned} \tag{3.12}$$

missä jälkimmäisin yhtäsuuruus seuraa $\boldsymbol{Q}_{k+1}$:n sarakkaiden ortonormaalisuudesta. Jäljelle on jäänyt pienimmän neliösumman tehtävä. On löydettävä $\boldsymbol{z}_k \in \mathbb{C}^k$ siten,

että $\left\| \|\boldsymbol{v}\|\boldsymbol{e}_1 - \widehat{\boldsymbol{H}}_k \boldsymbol{z}_k \right\|$ on pienin. Tässä $\widehat{\boldsymbol{H}}_k = [h_{ij}]$ on $(k+1) \times k$-matriisi, jonka alkiot ovat nollia $h_{ij} = 0$, kun $i > j + 1$.

Tehtävä voidaan ratkaista laskemalla QR-hajotelma Givensin rotaatioilla. Olkoon $\boldsymbol{G} \in \mathbb{C}^{(k+1)\times(k+1)}$ laskettu ortonormaali matriisi siten, että $\boldsymbol{G}\widehat{\boldsymbol{H}}_k = \widehat{\boldsymbol{R}}_k$ on yläkolmiomatriisi. Merkitään $\widehat{\boldsymbol{\eta}}_k = \boldsymbol{G}\boldsymbol{e}_1$ ja olkoon vektori $\boldsymbol{\eta}_k$ tämän k ensimmäistä komponenttia ja ζ_k viimeinen komponentti, jolloin $\widehat{\boldsymbol{\eta}}_k = \begin{bmatrix} \boldsymbol{\eta}_k \\ \zeta_k \end{bmatrix}$. Merkitään vielä $\boldsymbol{R}_k$:lla sitä $k \times k$-matriisia, joka saadaan poistamalla $\widehat{\boldsymbol{R}}_k$:sta viimeinen (nolla)rivi.

Käyttäen $\boldsymbol{G}$:n ortonormaalisuutta, tulee (3.12) muotoon

$$\begin{aligned} \|\boldsymbol{r}_k\|^2 &= \left\| \|\boldsymbol{v}\|\boldsymbol{e}_1 - \widehat{\boldsymbol{H}}_k \boldsymbol{z}_k \right\|^2 = \left\| \boldsymbol{G}(\|\boldsymbol{v}\|\boldsymbol{e}_1 - \widehat{\boldsymbol{H}}_k \boldsymbol{z}_k) \right\|^2 \\ &= \left\| \|\boldsymbol{v}\|\widehat{\boldsymbol{\eta}}_k - \widehat{\boldsymbol{R}}_k \boldsymbol{z}_k \right\|^2 = \left\| \|\boldsymbol{v}\|\boldsymbol{\eta}_k - \boldsymbol{R}_k \boldsymbol{z}_k \right\|^2 + \|\boldsymbol{v}\|^2 |\zeta_k|^2. \end{aligned} \tag{3.13}$$

Kun Arnoldin menetelmä ei ole vielä murtunut, on $\widehat{\boldsymbol{H}}_k$:n rangi määritteen (3.8) perusteella k. Kun $\boldsymbol{A}$ on kääntyvä, on $\widehat{\boldsymbol{H}}_k$:n rangi k myös murtumisaskeleella yhtälöstä (3.10) johtuen. Näissä tapauksissa $\widehat{\boldsymbol{R}}_k$:n ja $\boldsymbol{R}_k$:n rangi on myös k, joten $\boldsymbol{R}_k$ on kääntyvä. Ratkaistaan yhtälö

$$\boldsymbol{R}_k \boldsymbol{z}_k = \|\boldsymbol{v}\|\boldsymbol{\eta}_k$$

taaksepäin sijoituksin, jonka jälkeen GMRES-menetelmän likimääräisratkaisu saadaan kaavasta $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{Q}_k \boldsymbol{z}_k$. Tällöin $\|\boldsymbol{r}_k\| = \|\boldsymbol{v}\||\zeta_k|$.

Huomataan lisäksi, että Arnoldin menetelmän murtuessa $h_{k+1,k} = 0 \quad (\boldsymbol{w}_k = 0)$, jolloin GMRES antaa tarkan ratkaisun yhtälölle. Tällöin $\boldsymbol{G}_k$ on yksikkömatriisi, joten $\zeta_k = 0$ ja kaavasta (3.13) nähdään $\boldsymbol{r}_k = 0$.

Seuraava lause antaa mahdollisuuden arvioida GMRESin suppenemisnopeutta.

Lause 3.4.1. *Olkoon $\boldsymbol{A}$ diagonalisoituva, $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$. Tällöin jäännökselle pätee arvio*

$$\|\boldsymbol{r}_k\| \leq \|\boldsymbol{r}_0\| \kappa_2(\boldsymbol{X}) \min_{p \in \mathbb{P}_k, p(0)=1} \max_{\lambda \in \sigma(\boldsymbol{A})} |p(\lambda)|,$$

missä $\mathbb{P}_k$ on korkeintaan k-asteisten kompleksikertoimisten polynomien joukko ja $\kappa_2(\boldsymbol{X}) = \|\boldsymbol{X}\|_2 \|\boldsymbol{X}^{-1}\|_2$ on häiriöalttius 2-normissa.

Todistus. Olkoon p jokin korkeintaan k-asteinen polynomi siten, että $p(0) = 1$. Asetetaan $q(\lambda) = (1 - p(\lambda))/\lambda$ ja $\boldsymbol{x} = \boldsymbol{x}_0 + q(\boldsymbol{A})\boldsymbol{r}_0$. Tällöin

$$\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0 - (\boldsymbol{I} - p(\boldsymbol{A}))\boldsymbol{r}_0 = p(\boldsymbol{A})\boldsymbol{r}_0.$$

Koska q on $(k-1)$-asteinen polynomi, kuuluu $\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$. GMRES minimoi jäännöksen normin $\|\boldsymbol{r}_k\|$ avaruudessa $\boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, joten

$$\begin{aligned} \|\boldsymbol{r}_k\| &\leq \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\| = \|p(\boldsymbol{A})\boldsymbol{r}_0\| = \|\boldsymbol{X}p(\boldsymbol{\Lambda})\boldsymbol{X}^{-1}\boldsymbol{r}_0\| \\ &\leq \|\boldsymbol{X}\|\|p(\boldsymbol{\Lambda})\|\|\boldsymbol{X}^{-1}\|\|\boldsymbol{r}_0\| = \|\boldsymbol{r}_0\|\kappa_2(\boldsymbol{X}) \max_{\lambda \in \sigma(\boldsymbol{A})} |p(\lambda)|, \end{aligned}$$

koska diagonaalimatriisin 2-normi on itseisarvoltaan suurin diagonaalialkio. On lisäksi helppo nähdä, että oikeanpuolimmaisen lausekkeen minimi saavutetaan jollakin polynomilla p. □

Algoritmi 3.4.1 GMRES-menetelmä yhtälön $\boldsymbol{Ax} = \boldsymbol{b}$ likimääräiseksi ratkaisemiseksi. Suorittaa k iteraatiota tai pysähtyy, jos tarkka ratkaisu saavutetaan.

$\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{q}_1 \leftarrow \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$, $\widehat{\boldsymbol{\eta}} \leftarrow [1] \in \mathbb{C}^{1\times 1}$
for $j = 1$ to k **do**
  $\boldsymbol{w} \leftarrow \boldsymbol{A}\boldsymbol{q}_j$
  for $i = 1$ to j **do**
    $h_{ij} \leftarrow \langle \boldsymbol{w}, \boldsymbol{q}_i \rangle$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - h_{ij}\boldsymbol{q}_i$
  end for
  $h_{j+1,j} \leftarrow \|\boldsymbol{w}\|$
  $\boldsymbol{s} \leftarrow \begin{bmatrix} h_{1j} & h_{2j} & \cdots & h_{j+1,j} \end{bmatrix}^T$
  for $i = 1$ to $j - 1$ **do**
    $\boldsymbol{s}_{i:(i+1)} \leftarrow \boldsymbol{G}_i \boldsymbol{s}_{i:(i+1)}$
  end for
  Olkoon $\boldsymbol{G}_j \in \mathbb{C}^{2\times 2}$ Givensin rotaatio siten, että $\boldsymbol{G}_j \boldsymbol{s}_{j:(j+1)} = \begin{bmatrix} \# & 0 \end{bmatrix}^T$
  $\boldsymbol{s}_{j:(j+1)} \leftarrow \boldsymbol{G}_j \boldsymbol{s}_{j:(j+1)}$
  $\widehat{\boldsymbol{R}} \leftarrow \begin{bmatrix} \widehat{\boldsymbol{R}} \\ 0 \end{bmatrix} \in \mathbb{C}^{(j+1)\times(j-1)}$, $\widehat{\boldsymbol{R}} \leftarrow \begin{bmatrix} \widehat{\boldsymbol{R}} & \boldsymbol{s} \end{bmatrix}$, $\widehat{\boldsymbol{\eta}} \leftarrow \begin{bmatrix} \widehat{\boldsymbol{\eta}} \\ 0 \end{bmatrix}$, $\widehat{\boldsymbol{\eta}}_{j:(j+1)} \leftarrow \boldsymbol{G}_j \widehat{\boldsymbol{\eta}}_{j:(j+1)}$
  if $h_{j+1,j} = 0$ **then**
    break
  end if
  $\boldsymbol{q}_{j+1} \leftarrow \boldsymbol{w}/h_{j+1,j}$
end for
$\boldsymbol{Q} \leftarrow [\boldsymbol{q}_1 \cdots \boldsymbol{q}_j]$, $\boldsymbol{R} \leftarrow \widehat{\boldsymbol{R}}_{1:j,*}$, $\boldsymbol{\eta} \leftarrow \widehat{\boldsymbol{\eta}}_{1:j}$
Ratkaistaan $\boldsymbol{z}$ yhtälöstä $\boldsymbol{Rz} = \|\boldsymbol{r}_0\|\boldsymbol{\eta}$
$\boldsymbol{x} \leftarrow \boldsymbol{x}_0 + \boldsymbol{Q}\boldsymbol{z}$, jäännöksen normi on $\|\boldsymbol{r}_0\||\widehat{\boldsymbol{\eta}}_{j+1}|$

Seuraus 3.4.2. *Olkoon $\boldsymbol{A}$ diagonalisoituva ja kuulukoon sen ominaisarvot ζ-keskiseen R-säteiseen suljettuun kiekkoon. Oletetaan lisäksi, että origo ei kuulu tähän kiekkoon. Tällöin*

$$\|\boldsymbol{r}_k\| \leq \|\boldsymbol{r}_0\| \kappa_2(\boldsymbol{X}) \left(\frac{R}{|\zeta|}\right)^k.$$

Todistus. Olkoon polynomi $p(\lambda) = (1 - \lambda/\zeta)^k$. Saadaan

$$|p(\lambda)| = \left(\frac{1}{|\zeta|}|\zeta - \lambda|\right)^k \leq \left(\frac{R}{|\zeta|}\right)^k.$$

□

Edellisen lauseen ja sen seurauksen sisältämä häiriöalttius $\kappa_2(\boldsymbol{X})$ on yleensä vaikea laskea, mikä tekee arvioiden käyttämisestä hankalaa. Kuitenkin normaalimatriiseille (tai lähes normaaleille) pätee $\kappa_2(\boldsymbol{X}) = 1$, koska matriisi on normaali jos ja vain jos se on diagonalisoituva ortonormeeratulla muunnosmatriisilla $\boldsymbol{X}$.

3.4.1 Uudelleenkäynnistys

GMRESin suorittama täysi ortogonalisointi vaatii kaikkien vektorien $\boldsymbol{q}_j \in \mathbb{C}^n$ tallentamisen tietokoneen muistiin. Vaadittu muistin määrä kierrosten k kasvaessa on siten $O(nk)$, jolloin tietokoneen muistikapasiteetin rajat saattavat tulla vastaan laskennan edistyessä. Jokaisen kierroksen laskenta myös hidastuu, koska uusi vektori täytyy ortogonalisoida kaikkia aikaisempia vastaan. Näistä syistä voi olla edullista käynnistää algoritmi uudelleen käyttäen alkuarvauksena $\boldsymbol{x}_0$ edellisen algoritmin ajon likimääräisratkaisua $\boldsymbol{x}_k$. Merkitään k:n iteraatiokierroksen välein uudelleenkäynnistettyä menetelmää GMRES(k):lla.

Lause 3.4.3. *Jos* $\langle \boldsymbol{Ax}, \boldsymbol{x} \rangle \neq 0$ *kaikilla* $\boldsymbol{x} \neq 0$, *niin GMRES(k) suppenee.*

Todistus. Olkoon $\boldsymbol{x}_0$ alkuarvaus ja $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Ax}_0$. Olkoon $\boldsymbol{x}_1$ ensimmäinen GMRESin antama likimääräisratkaisu eli $\|\boldsymbol{b} - \boldsymbol{Ax}\|$ $(\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_1(\boldsymbol{A}, \boldsymbol{r}_0))$ on minimissään, kun $\boldsymbol{x} = \boldsymbol{x}_1$. Tällöin $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \alpha \boldsymbol{r}_0$ ja jäännös $\boldsymbol{r}_1 = \boldsymbol{b} - \boldsymbol{Ax}_1 = \boldsymbol{r}_0 - \alpha \boldsymbol{Ar}_0$. Normille

$$\|\boldsymbol{r}_1\|^2 = \langle \boldsymbol{r}_0 - \alpha \boldsymbol{Ar}_0, \boldsymbol{r}_0 - \alpha \boldsymbol{Ar}_0 \rangle = \langle \boldsymbol{r}_0 - \alpha \boldsymbol{Ar}_0, \boldsymbol{r}_0 \rangle - \overline{\alpha} \langle \boldsymbol{r}_0 - \alpha \boldsymbol{Ar}_0, \boldsymbol{Ar}_0 \rangle , \quad (3.14)$$

missä $\langle \boldsymbol{r}_0 - \alpha \boldsymbol{Ar}_0, \boldsymbol{Ar}_0 \rangle = 0$, koska $\boldsymbol{r}_1 \perp \boldsymbol{A}\mathcal{K}_1(\boldsymbol{A}, \boldsymbol{r}_0)$. Tästä saadaan myös

$$\alpha = \frac{\langle \boldsymbol{r}_0, \boldsymbol{Ar}_0 \rangle}{\langle \boldsymbol{Ar}_0, \boldsymbol{Ar}_0 \rangle}.$$

Jatkaen kohdasta (3.14)

$$\|\boldsymbol{r}_1\|^2 = \|\boldsymbol{r}_0\|^2 - \alpha \langle \boldsymbol{Ar}_0, \boldsymbol{r}_0 \rangle = \|\boldsymbol{r}_0\|^2 \left(1 - \frac{|\langle \boldsymbol{Ar}_0, \boldsymbol{r}_0 \rangle|^2 \langle \boldsymbol{r}_0, \boldsymbol{r}_0 \rangle}{\langle \boldsymbol{r}_0, \boldsymbol{r}_0 \rangle^2 \langle \boldsymbol{Ar}_0, \boldsymbol{Ar}_0 \rangle} \right).$$

Asetetaan $\mu = \min_{\|\boldsymbol{x}\|=1} |\langle \boldsymbol{Ax}, \boldsymbol{x} \rangle|$ ja käytetään epäyhtälöä $\|\boldsymbol{Ar}_0\| \leq \|\boldsymbol{A}\| \|\boldsymbol{r}_0\|$, jolloin

$$\|\boldsymbol{r}_1\|^2 \leq \|\boldsymbol{r}_0\|^2 \left(1 - \frac{\mu^2}{\|\boldsymbol{A}\|^2} \right).$$

Täten GMRESin ensimmäinen iteraatio pienentää jäännöksen normia $\|\boldsymbol{r}_0\|$ ainakin tekijällä $(1 - \mu^2/\|\boldsymbol{A}\|^2)^{1/2} < 1$. Muissa iteraatioissa jäännöksen normi ei kasva. Seuraavissakin uudelleenkäynnistyksissä ensimmäinen iteraatio aina pienentää normia samalla tekijällä. Näin ollen uudelleenkäynnistetty GMRES suppenee. □

3.5 $\mathbb{R}$-lineaariset menetelmät

Olkoon $\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^n$, $\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{Mz} + \boldsymbol{M}_{\#}\bar{\boldsymbol{z}}$ reaalilineaarinen operaattori, $\boldsymbol{b} \in \mathbb{C}^n$ ja tuntematon vektori $\boldsymbol{z}$ on ratkaistava yhtälöstä

$$\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{b}. \quad (3.15)$$

Tällainen yhtälö voitaisiin ratkaista muodostamalla reaalimatriisiesitys ja käyttämällä edellä kuvattuja menettelyjä matriisiyhtälöiden ratkaisemiseksi. Pitämällä $\mathbb{R}$-lineaarinen yhtälö annetussa muodossaan tarjoaa kuitenkin etuja. Etsittäessä reaalimatriisiesityksen likimääräisratkaisua jostakin k:n vektorin virittämästä $\mathbb{R}^{2n}$:n aliavaruudesta tilanne voidaan siirtää alkuperäisen yhtälön (3.15) ratkaisemiseen.

Viritettyä aliavaruutta vastaa tällöin k:n vektorin virittämä $\mathbb{R}$-kertoiminen $\mathbb{C}^n$:n aliavaruus. Tässä kappaleessa esitellään menetelmiä, joissa ratkaisua etsitään näiden k:n vektorin virittämästä $\mathbb{C}$-*kertoimisesta* aliavaruudesta. Likimääräisratkaisun hakuavaruus kullakin iteraatiolla on siten laajempi ja pyrkimyksenä on saavuttaa reaalimatriisimenetelmiä nopeampi suppeneminen.

Yleisen reaalilineaarisen operaattori sijaan keskitytään pääasiassa muotoa $\mathcal{M}_\kappa(\boldsymbol{z}) = \kappa\boldsymbol{z} + \boldsymbol{M}_\#\bar{\boldsymbol{z}}$ ($\kappa \in \mathbb{C}$) oleviin operaattoreihin, koska näiden ominaisuudet soveltuvat Krylovin aliavaruuksien käyttöön. Tämän muotoisilla operaattoreilla on sovelluksia mm. lääketieteellisessä kuvantamisessa. Sähköisessä impedanssitomografiassa muodostetaan poikkileikkauskuva esimerkiksi potilaan rintakehästä syöttämällä ihon pinnalle sähkövirtoja ja mittaamalla vasteena saatu sähköpotentiaali. Kyseessä on inversio-ongelma, jonka ratkaisuna on sähkönjohtavuus poikkileikkauksen jokaisessa pisteessä. Kuva keuhkoista ja sydämestä voidaan muodostaa, koska eri kudoksilla on erilaiset johtavuudet. Ongelmaa ratkaistaessa joudutaan heikosti singulaarisiin Fredholmin toisen lajin integraaliyhtälöihin, jolloin diskretoinnin myötä päädytään yhtälöihin muotoa $\boldsymbol{z} + \boldsymbol{M}_\#\bar{\boldsymbol{z}} = \mathbf{1}$, missä $\mathbf{1}$ on luvuista 1 koostuva vektori. Tässä matriisia $\boldsymbol{M}_\#$ ei tallenneta tietokoneen muistiin vaan operaatio $\boldsymbol{z} \to \boldsymbol{M}_\#\bar{\boldsymbol{z}}$ on peräisin funktioiden konvoluutiosta ja diskretoitu operaatio voidaan laskea käyttäen nopeaa Fourier-muunnosta (FFT). Yhtälöt ovat yleensä vähintään kokoa $n = 2^{16}$ ja yhden kuvan laskemiseen sellainen joudutaan ratkaisemaan jokaisessa diskretointipisteessä. Suorat ratkaisumenetelmät eivät tule kyseeseen ja iteratiivisissa menetelmissä suppenemisnopeuden parantaminen antaa kuvan nopeammin lääkärin käyttöön.

Muotoillaan seuraavaksi Petrov-Galerkinin yhtälö matriiseja vastaavalla tavalla. Olkoon X_k jokin $\mathbb{C}^n$:n $\mathbb{C}$-kertoiminen aliavaruus ja Y_k jokin $\mathbb{R}$-kertoiminen aliavaruus. Kun $\boldsymbol{z}_0 \in \mathbb{C}^n$ on ratkaisun alkuarvaus, Petrov-Galerkinin ehto likimääräisratkaisuksi $\boldsymbol{z}_k \in \boldsymbol{z}_0 + X_k$ on

$$\langle \boldsymbol{b} - \mathcal{M}(\boldsymbol{z}_k), \boldsymbol{y} \rangle = 0 \qquad \text{kaikilla } \boldsymbol{y} \in Y_k. \tag{3.16}$$

Merkitään $\boldsymbol{r}_0 = \boldsymbol{b} - \mathcal{M}(\boldsymbol{z}_0)$. Kun matriisien $\boldsymbol{V}_k$ ja $\boldsymbol{W}_k$ sarakkeet muodostavat kannat avaruuksille X_k ja Y_k (vastaavasti), niin edellä oleva ehto voidaan kirjoittaa muotoon

$$\boldsymbol{W}_k^*\boldsymbol{M}\boldsymbol{V}_k\boldsymbol{u}_k + \boldsymbol{W}_k^*\boldsymbol{M}_\#\overline{\boldsymbol{V}}_k\overline{\boldsymbol{u}}_k = \boldsymbol{W}_k^*\boldsymbol{r}_0\,, \tag{3.17}$$

mistä ratkaisemalla $\boldsymbol{u}_k$ saadaan likimääräisratkaisu alkuperäiselle yhtälölle (3.15) laskemalla

$$\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{V}_k\boldsymbol{u}_k. \tag{3.18}$$

3.6 Krylovin aliavaruudet

Krylovin aliavaruudet voidaan määritellä matriiseja vastaavalla tavalla. Olkoon

$$\mathcal{K}_k(\mathcal{M}, \boldsymbol{r}_0) = \operatorname{span}\left\{\boldsymbol{r}_0, \mathcal{M}(\boldsymbol{r}_0), \mathcal{M}^2(\boldsymbol{r}_0), \ldots, \mathcal{M}^{k-1}(\boldsymbol{r}_0)\right\} \qquad (k = 1, 2, \ldots).$$

Vektorin $\boldsymbol{r}_0$ minimipolynomi $\mathcal{M}$:n suhteen on pienintä astetta oleva kompleksikertoiminen mooninen polynomi p siten, että $p(\mathcal{M})(\boldsymbol{r}_0) = 0$. Kun operaattorille $\mathcal{M}$ muodostetaan sitä vastaava reaalinen $2n \times 2n$-matriisi $\boldsymbol{A}$, niin Cayley-Hamiltonin lauseen mukaan on olemassa (reaalikertoiminen) polynomi p siten, että $p(\boldsymbol{A}) = 0$. Tällä samalla polynomilla myös $p(\mathcal{M}) = 0$, joten minimipolynomi on hyvin määritelty.

Seuraavassa keskitytään siirrettyihin antilineaarisiin operaattoreihin $\mathcal{M}_\kappa(z) = \kappa z + M_\# \overline{z}$, koska yleisemmillä reaalilineaarisilla operaattoreilla ei ole tarvittavia ominaisuuksia. Kuitenkin mikäli operaattorin $\mathcal{M}(z) = Mz + M_\# \overline{z}$ matriisi M on helposti kääntyvä, niin yhtälö $\mathcal{M}(z) = b$ voidaan kertoa puolittain M^{-1}:llä ja siten saadaan operaattori muotoon $\widetilde{\mathcal{M}}(z) = z + M^{-1} M_\# \overline{z}$.

Lemma 3.6.1. *Olkoon $\mathcal{M}_\kappa(z) = \kappa z + M_\# \overline{z}$ siirretty antilineaarinen operaattori ja k ei-negatiivinen kokonaisluku. Tällöin*

$$\mathcal{M}_\kappa^k = p(\mathcal{M}_0), \tag{3.19}$$

$$\mathcal{M}_0^k = q(\mathcal{M}_\kappa), \tag{3.20}$$

joillakin k-asteisilla moonisilla polynomeilla p ja q. Lisäksi, kun $\tilde{p}$ on k-asteinen polynomi, niin

$$\mathcal{M}_\kappa \tilde{p}(\mathcal{M}_\kappa) = \tilde{q}(\mathcal{M}_\kappa), \tag{3.21}$$

jollakin $(k+1)$-asteisella polynomilla $\tilde{q}$.

Todistus. Kirjoitetaan ensimmäistä kaavaa varten

$$\mathcal{M}_\kappa^k = (\kappa I + \mathcal{M}_0)^k = (\kappa I + \mathcal{M}_0) \cdots (\kappa I + \mathcal{M}_0).$$

Kerrotaan tämä auki ja käytetään kaavoja $\mathcal{M}_0 \circ \kappa I = \overline{\kappa} \mathcal{M}_0$ ja $\kappa I \circ \mathcal{M}_0 = \kappa \mathcal{M}_0$, niin saadaan (3.19).

Jo osoitetun kaavan perusteella on olemassa korkeintaan $(k-1)$-asteinen polynomi r siten, että $\mathcal{M}_0^k = \mathcal{M}_\kappa^k - r(\mathcal{M}_0)$. Kaava (3.20) seuraa siten induktiosta, koska se on selvästi tosi $k = 0$:lla.

Olkoon vielä $\tilde{p}$ jokin k-asteinen polynomi. Tällöin (3.19) perusteella on olemassa k-asteinen polynomi $\tilde{r}$ siten, että

$$\mathcal{M}_\kappa \tilde{p}(\mathcal{M}_\kappa) = \mathcal{M}_\kappa \tilde{r}(\mathcal{M}_0) = \kappa \tilde{r}(\mathcal{M}_0) + \mathcal{M}_0 \tilde{r}(\mathcal{M}_0) = \kappa \tilde{r}(\mathcal{M}_0) + \tilde{s}(\mathcal{M}_0),$$

jollakin $(k+1)$-asteisella polynomilla $\tilde{s}$, koska $\mathcal{M}_0 \circ \alpha_j \mathcal{M}_0^j = \overline{\alpha_j} \mathcal{M}_0^{j+1}$. Käyttämällä vielä kaavaa (3.20), saadaan viimeinen väite. □

Huomautus 3.6.2. Lemman perusteella erityisesti $\mathcal{K}_k(\mathcal{M}_\kappa, r_0) = \mathcal{K}_k(\mathcal{M}_0, r_0)$. Antilineaariselle operaattorille $\mathcal{M}_0(z) = M_\# \overline{z}$ ja vektorille $v \in \mathbb{C}^n$ pätee

$$\mathcal{M}_0^0(v) = v,$$

$$\mathcal{M}_0^{k+1}(v) = M_\# \overline{\mathcal{M}_0^k(v)}.$$

Tällöin esimerkiksi $\mathcal{M}_0(v) = v$, $\mathcal{M}_0^1(v) = M_\# \overline{v}$, $\mathcal{M}_0^2(v) = M_\# \overline{M_\#} v$, ja $\mathcal{M}_0^3(v) = M_\# \overline{M_\#} M_\# \overline{v}$. Täten

$$\mathcal{K}_k(\mathcal{M}_\kappa, r_0) = \operatorname{span}\left\{ r_0, M_\# \overline{r}_0, M_\# \overline{M_\#} r_0, \ldots, \mathcal{M}_0^{k-1}(r_0) \right\}.$$

Seuraava lause siirretyille antilineaarisille operaattoreille vastaa matriisien lausetta 3.3.1. Lause ei yleisty kaikille reaalilineaarisille operaattoreille.

Lause 3.6.3. *Olkoot $\mathcal{M}_\kappa$ siirretty antilineaarinen operaattori, $r_0 \in \mathbb{C}^n$ ja p on r_0:n minimipolynomi. Kun merkitään $\mu = \deg p$, niin*

(1) $\dim \mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0) = k$, *kun* $k \leq \mu$,
(2) $\mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0) = \mathcal{K}_\mu(\mathcal{M}_\kappa, \boldsymbol{r}_0)$, *kun* $k \geq \mu$.

Lisäksi $\mathcal{K} = \mathcal{K}_\mu(\mathcal{M}_\kappa, \boldsymbol{r}_0)$ *on* $\mathcal{M}_\kappa$*:n invariantti aliavaruus,* $\mathcal{M}_\kappa(\mathcal{K}) \subset \mathcal{K}$.

Todistus. Ensimmäisen kohdan todistus on sama kuin lauseessa 3.3.1. Vektorin $\boldsymbol{r}_0$ minimipolynomin perusteella $\mathcal{M}_\kappa^\mu(\boldsymbol{r}_0) \in \mathcal{K}_\mu(\mathcal{M}_\kappa, \boldsymbol{r}_0)$, mistä saadaan $(\mu-1)$-asteinen polynomi $\tilde{p}$ siten, että

$$\mathcal{M}_\kappa^\mu(\boldsymbol{r}_0) = \tilde{p}(\mathcal{M}_\kappa)(\boldsymbol{r}_0). \tag{3.22}$$

Nyt $\mathcal{M}_\kappa^{\mu+1}(\boldsymbol{r}_0) = \mathcal{M}_\kappa(\mathcal{M}_\kappa^\mu(\boldsymbol{r}_0)) = \mathcal{M}_\kappa(\tilde{p}(\mathcal{M}_\kappa)(\boldsymbol{r}_0))$. Kohdan (3.21) perusteella on olemassa μ-asteinen polynomi $\tilde{q}$ siten, että $\mathcal{M}_\kappa^{\mu+1}(\boldsymbol{r}_0) = \tilde{q}(\mathcal{M}_\kappa)(\boldsymbol{r}_0)$. Käyttämällä vielä kaavaa (3.22), saadaan $\mathcal{M}_\kappa^{\mu+1}(\boldsymbol{r}_0) \in \mathcal{K}_\mu(\mathcal{M}_\kappa, \boldsymbol{r}_0)$.

Koska $\mathcal{M}_\kappa^{\mu+2}(\boldsymbol{r}_0) = \mathcal{M}_\kappa(\mathcal{M}_\kappa^{\mu+1}(\boldsymbol{r}_0))$, voidaan edellistä jatkaa ja kohdan 2 väite seuraa induktiosta. Avaruuden $\mathcal{K}_\mu(\mathcal{M}_\kappa, \boldsymbol{r}_0)$ invarianssi on selvä edeltävän perusteella. □

Ryhdytään sitten laskemaan ortonormaaleja vektoreita noudattamalla matriisien Arnoldin menetelmää. Valitaan $\boldsymbol{q}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$. Kun kantavektorit $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_j$ on laskettu, niin $\boldsymbol{q}_{j+1}$ saadaan laskemalla $\mathcal{M}(\boldsymbol{q}_j)$ ja ortogonalisoimalla se käyttäen Gram-Schmidt menetelmää

$$h_{ij} = \langle \mathcal{M}(\boldsymbol{q}_j), \boldsymbol{q}_i \rangle \qquad (i = 1, 2, \ldots, j) \tag{3.23}$$

$$\boldsymbol{w}_j = \mathcal{M}(\boldsymbol{q}_j) - \sum_{i=1}^{j} h_{ij}\boldsymbol{q}_i \tag{3.24}$$

$$h_{j+1,j} = \|\boldsymbol{w}_j\| \tag{3.25}$$

Menetelmä pysäytetään, jos $\boldsymbol{w}_j = 0$. Muutoin asetetaan $\boldsymbol{q}_{j+1} = \boldsymbol{w}_j/\|\boldsymbol{w}_j\|$.

On huomattava, että sisätulon arvo kaavassa (3.23) on yleensä kompleksiluku. Reaalilineaarisille operaattoreille pätee vain $\mathcal{M}(\alpha \boldsymbol{z}) = \alpha \mathcal{M}(\boldsymbol{z})$, missä α on reaalinen. Tästä syystä vektori $\boldsymbol{w}_j$ ei välttämättä enää kuulu Krylovin aliavaruuteen eikä algoritmi siten generoi sille kantaa. Operaattoreille $\mathcal{M}_\kappa$ pätee kuitenkin matriiseja vastaavat lauseet.

Lause 3.6.4. *Oletetaan, että operaattorilla* $\mathcal{M}_\kappa$ *on Arnoldin menetelmällä laskettu vektorit* $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$. *Nämä vektorit muodostavat ortonormaalin kannan avaruuteen* $\mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0)$.

Todistus. Vektorit ovat ortonormaalit Gram-Schmidt prosessin ja normeerauksen myötä. Osoitetaan, että jokaiselle $\boldsymbol{q}_j$ on olemassa astetta $j-1$ oleva polynomi p_{j-1} siten, että $\boldsymbol{q}_j = p_{j-1}(\mathcal{M}_\kappa)(\boldsymbol{r}_0)$. Tämä on selvästi totta $\boldsymbol{q}_1$:lle. Oletetaan sitten, että näin on $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_j$:lle ja osoitetaan todeksi myös $\boldsymbol{q}_{j+1}$:lle. Kaavasta (3.24) saadaan

$$\begin{aligned}
\boldsymbol{q}_{j+1} &= \tfrac{1}{\|\boldsymbol{w}_j\|}\mathcal{M}_\kappa(\boldsymbol{q}_j) - \sum_{i=1}^{j} \tfrac{1}{\|\boldsymbol{w}_j\|} h_{ij}\boldsymbol{q}_i \\
&= \tfrac{1}{\|\boldsymbol{w}_j\|}\mathcal{M}_\kappa p_{j-1}(\mathcal{M}_\kappa)(\boldsymbol{r}_0) - \sum_{i=1}^{j} \tfrac{1}{\|\boldsymbol{w}_j\|} h_{ij} p_{i-1}(\mathcal{M}_\kappa)(\boldsymbol{r}_0),
\end{aligned}$$

missä $\mathcal{M}_\kappa p_{j-1}(\mathcal{M}_\kappa) = q(\mathcal{M}_\kappa)$ jollakin j-asteisella polynomilla lemman 3.6.1 mukaan. Täten oikealla puolella on j-asteinen polynomi, joten induktioaskel on osoitettu.

Ollaan saatu, että $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$ kuuluvat avaruuteen $\mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0)$. Selvästi aina $\dim \mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0) \leq k$, joten vektorit virittävät avaruuden ja $\dim \mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0) = k$. □

Lause 3.6.5. *Operaattorilla $\mathcal{M}_\kappa$ Arnoldin menetelmässä $\boldsymbol{w}_j = 0$ jos ja vain jos j on $\boldsymbol{v}$:n minimipolynomin asteluku.*

Todistus. Oletetaan aluksi $\boldsymbol{w}_j = 0$. Kuten lauseen 3.6.4 todistuksessa, olkoon p_{j-1} astetta j oleva polynomi siten, että $\boldsymbol{q}_j = p_{j-1}(\mathcal{M}_\kappa)(\boldsymbol{r}_0)$. Kaavasta (3.24) saadaan

$$0 = \boldsymbol{w}_j = \mathcal{M}_\kappa(\boldsymbol{q}_j) - \sum_{i=1}^{j} h_{ij}\boldsymbol{q}_i = \mathcal{M}_\kappa p_{j-1}(\mathcal{M}_\kappa)(\boldsymbol{r}_0) - \sum_{i=1}^{j} h_{ij} p_{i-1}(\mathcal{M}_\kappa)(\boldsymbol{r}_0),$$

jonka oikealla puolella on j-asteinen polynomi lemman 3.6.1 perusteella. Siispä $\boldsymbol{r}_0$:n minimipolynomin asteluku on korkeintaan j. Koska $\boldsymbol{w}_{j-1} \neq 0$, on toisaalta saatu muodostettua $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_j$, joten lauseen 3.6.4 perusteella $\dim \mathcal{K}_j(\mathcal{M}_\kappa, \boldsymbol{r}_0) = j$ ja lauseen 3.6.3 mukaan minimipolynomin asteluku ei voi olla pienempi kuin j.

Oletetaan sitten, että j on $\boldsymbol{r}_0$:n minimipolynomin asteluku. Lauseiden 3.6.3 ja 3.6.4 perusteella Arnoldin menetelmällä on mahdotonta muodostaa enemmän kuin j ortonormaalia vektoria. Täytyy siis olla $\boldsymbol{w}_i = 0$ jollakin $i \leq j$. Toisaalta, jos $i < j$, niin tämän todistuksen ensimmäisen osan mukaan asteluvunkin täytyisi olla i. Täten $\boldsymbol{w}_j = 0$. □

Huomautuksen 3.6.2 mukaisesti operaattorin $\mathcal{M}_\kappa$ Krylovin aliavaruudet ovat samoja kuin $\mathcal{M}_0$:n, joten yksinkertaisuuden vuoksi muodostetaan kantavektorit käyttäen $\mathcal{M}_0$:aa. Olkoot $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$ saatu operaattorilla $\mathcal{M}_0$ Arnoldin menetelmästä ja asetetaan $\boldsymbol{Q}_k = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_k \end{bmatrix}$. Olkoon $\widehat{\boldsymbol{H}}_k$ Arnoldin menetelmän alkioista h_{ij} muodostettu $(k+1) \times k$-matriisi, $(\widehat{\boldsymbol{H}}_k)_{ij} = h_{ij} \quad (i \leq j+1)$ ja muut alkiot nollia. Olkoon vielä $\boldsymbol{H}_k$ se $k \times k$-matriisi, kun matriisista $\widehat{\boldsymbol{H}}_k$ poistetaan viimeinen rivi. Kaavasta (3.24) saadaan

$$h_{j+1,j}\boldsymbol{q}_{j+1} = \boldsymbol{M}_\# \overline{\boldsymbol{q}_j} - \sum_{i=1}^{j} h_{ij}\boldsymbol{q}_i \qquad \Leftrightarrow \qquad \boldsymbol{M}_\# \overline{\boldsymbol{q}_j} = \sum_{i=1}^{j+1} h_{ij}\boldsymbol{q}_i,$$

joka voidaan kirjoittaa matriisimuotoihin

$$\boldsymbol{M}_\# \overline{\boldsymbol{Q}}_j = \boldsymbol{Q}_{j+1}\widehat{\boldsymbol{H}}_j, \tag{3.26}$$

$$\boldsymbol{M}_\# \overline{\boldsymbol{Q}}_j = \boldsymbol{Q}_j \boldsymbol{H}_j + \boldsymbol{w}_j \boldsymbol{e}_j^T, \tag{3.27}$$

missä $\boldsymbol{e}_j$ on $\mathbb{C}^j$:n j:s luonnollinen kantavektori. Kertomalla jälkimmäinen puolittain $\boldsymbol{Q}_j^*$:llä, saadaan

$$\boldsymbol{Q}_j^* \boldsymbol{M}_\# \overline{\boldsymbol{Q}}_j = \boldsymbol{H}_j. \tag{3.28}$$

3.7 $\mathbb{R}$-lineaarinen täyden ortogonalisoinnin menetelmä

Olkoon $\boldsymbol{z}_0$ alkuarvaus yhtälön $\mathcal{M}_\kappa(\boldsymbol{z}) = \boldsymbol{b}$ ratkaisulle ja asetetaan $\boldsymbol{r}_0 = \boldsymbol{b} - \mathcal{M}_\kappa(\boldsymbol{z}_0)$. Täyden ortogonalisaation menetelmässä (Full Orthogonalization Method, FOM) lasketaan Arnoldin menetelmällä kantavektoreita Krylovin aliavaruuksiin $\mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0)$, jolloin likimääräisratkaisu $\boldsymbol{z}_k \in \boldsymbol{z}_0 + \mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0)$ saadaan kohtisuoruusvaatimuksesta

$$\langle \boldsymbol{b} - \mathcal{M}_\kappa(\boldsymbol{z}_k), \boldsymbol{y} \rangle = 0 \qquad \text{kaikilla } \boldsymbol{y} \in \mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0).$$

Kun matriisin $\boldsymbol{Q}_k$ sarakkeet ovat Arnoldin menetelmän ortonormaalit kantavektorit, yhtälöksi tulee

$$\kappa \boldsymbol{u}_k + \boldsymbol{Q}_k^* \boldsymbol{M}_\# \overline{\boldsymbol{Q}}_k \overline{\boldsymbol{u}}_k = \boldsymbol{Q}_k^* \boldsymbol{r}_0,$$

mistä ratkaisemalla $\boldsymbol{u}_k$ saadaan likimääräisratkaisuksi $\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{Q}_k \boldsymbol{u}_k$. Käyttäen yhtälöä (3.28) yksinkertaistuu tämä vielä muotoon

$$\kappa \boldsymbol{u}_k + \boldsymbol{H}_k \overline{\boldsymbol{u}_k} = \|\boldsymbol{r}_0\| \boldsymbol{e}_1, \tag{3.29}$$

$$\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{Q}_k \boldsymbol{u}_k. \tag{3.30}$$

Likimääräisratkaisun virheen arvioimiseksi käytetään edellä olevia yhtälöitä (3.27), (3.29) ja (3.30), jolloin

$$\begin{aligned} \boldsymbol{b} - \mathcal{M}_\kappa(\boldsymbol{z}_k) &= \boldsymbol{r}_0 - \mathcal{M}_\kappa(\boldsymbol{Q}_k \boldsymbol{u}_k) = \boldsymbol{r}_0 - \kappa \boldsymbol{Q}_k \boldsymbol{u}_k - \boldsymbol{M}_\# \overline{\boldsymbol{Q}}_k \overline{\boldsymbol{u}_k} \\ &= \boldsymbol{r}_0 - \kappa \boldsymbol{Q}_k \boldsymbol{u}_k - \boldsymbol{Q}_k \boldsymbol{H}_k \overline{\boldsymbol{u}_k} - (\boldsymbol{w}_k \boldsymbol{e}_k^T) \overline{\boldsymbol{u}_k} \\ &= \boldsymbol{Q}_k (\|\boldsymbol{r}_0\| \boldsymbol{e}_1 - \kappa \boldsymbol{u}_k - \boldsymbol{H}_k \overline{\boldsymbol{u}_k}) - \boldsymbol{w}_k (\boldsymbol{e}_k^T \overline{\boldsymbol{u}_k}) = -(\boldsymbol{e}_k^T \overline{\boldsymbol{u}_k}) \boldsymbol{w}_k, \end{aligned} \tag{3.31}$$

mistä edelleen

$$\boldsymbol{b} - \mathcal{M}_\kappa \boldsymbol{z}_k = -h_{k+1,k} (\boldsymbol{e}_k^T \overline{\boldsymbol{u}_k}) \boldsymbol{q}_{k+1}, \tag{3.32}$$

$$\|\boldsymbol{b} - \mathcal{M}_\kappa \boldsymbol{z}_k\| = h_{k+1,k} |\boldsymbol{e}_k^T \boldsymbol{u}_k|. \tag{3.33}$$

Tätä on mahdollista käyttää algoritmin pysäyttämiseksi, kun riittävä tarkkuus on saavutettu. Kaavasta 3.31 myös nähdään, että ratkaisu $\boldsymbol{z}_k$ on tarkka Arnoldin menetelmän murtuessa (kun $\boldsymbol{w}_k = 0$).

3.7.1 Menetelmä symmetriselle $\boldsymbol{M}_\#$

Symmetrisellä matriisilla $\boldsymbol{M}_\#$ saadaan edellä kuvattua FOMia yksinkertaistettua siten, että tietokoneen muistissa tarvitsee pitää ainoastaan kaksi viimeisintä kantavektori $\boldsymbol{q}_j$ ja likimääräisratkaisu saadaan eräänlaisen päivitysprosessin kautta.

Kun operaattorille $\mathcal{M}_\kappa$ ajetaan Arnoldin menetelmää symmetrisellä matriisilla $\boldsymbol{M}_\#$, niin yhtälöstä (3.28) nähdään

$$\boldsymbol{H}_k^T = (\boldsymbol{Q}_k^* \boldsymbol{M}_\# \overline{\boldsymbol{Q}}_k)^T = \overline{\boldsymbol{Q}}_k^T \boldsymbol{M}_\#{}^T (\boldsymbol{Q}_k^*)^T = \boldsymbol{Q}_k^* \boldsymbol{M}_\# \overline{\boldsymbol{Q}}_k = \boldsymbol{H}_k,$$

joten Hessenbergin matriisi $\boldsymbol{H}_k$ on myös symmetrinen. Täten se on kolmilävistäjämuotoa

$$\boldsymbol{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \beta_k & \alpha_k \end{bmatrix}.$$

Tällöin $\alpha_j = h_{jj}$ ja $\beta_j = h_{j,j-1}$ ja Arnoldin menetelmä (3.23)-(3.25) yksinkertaistuu seuraavaksi Lanczosin menetelmäksi. Asetetaan $\beta_1 = 0$, $\boldsymbol{q}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$ ja lasketaan vektorit $\boldsymbol{q}_2, \boldsymbol{q}_3, \ldots$ iteroimalla seuraavia kaavoja aloittaen arvosta $j = 1$.

$$\alpha_j = \langle \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_j, \boldsymbol{q}_j \rangle, \tag{3.34}$$

$$\boldsymbol{w}_j = \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_j - \alpha_j \boldsymbol{q}_j - \beta_j \boldsymbol{q}_{j-1}, \tag{3.35}$$

$$\beta_{j+1} = \|\boldsymbol{w}_j\|. \tag{3.36}$$

Menetelmä pysäytetään, jos $\beta_{j+1} = 0$. Muutoin asetetaan $\boldsymbol{q}_{j+1} = \boldsymbol{w}_j/\beta_{j+1}$. Muunnettua Gram-Schmidt menetelmää käyttäen tästä tulee algoritmi 3.7.1.

Algoritmi 3.7.1 Lanczosin menetelmä ortonormaalin kannan muodostamiseksi Krylovin aliavaruudelle $\mathcal{K}_k(\boldsymbol{\mathcal{M}}_0, \boldsymbol{r}_0)$ symmetrisellä $\boldsymbol{M}_{\#}$ käyttäen muunnettua Gram-Schmidt menetelmää. Laskee ortonormaalit kantavektorit $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots$. Algoritmi pysähtyy muodostettuaan $\min\{k+1, \dim \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})\}$ kantavektoria.

$\boldsymbol{q}_1 \leftarrow \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$, $\beta_1 \leftarrow 0$, $\boldsymbol{q}_0 \leftarrow 0$
for $j = 1$ to k **do**
  $\boldsymbol{w} \leftarrow \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_j - \beta_j \boldsymbol{q}_{j-1}$
  $\alpha_j \leftarrow \langle \boldsymbol{w}, \boldsymbol{q}_j \rangle$
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_j \boldsymbol{q}_j$
  $\beta_{j+1} \leftarrow \|\boldsymbol{w}\|$
  if $\beta_{j+1} = 0$ **then**
    lopeta algoritmi
  end if
  $\boldsymbol{q}_{j+1} \leftarrow \boldsymbol{w}/\beta_{j+1}$
end for

Lanczosin menetelmässä ortonormalisointiin tarvitaan vain kahta edellistä vektoria. FOMin likimääräisratkaisun laskemiseen kaavan (3.30) mukaan tarvitaan kuitenkin kaikkia vektoreita $\boldsymbol{q}_j$. Näytetään seuraavaksi miten tästä vaatimuksesta päästään eroon.

Merkitään $\boldsymbol{\mathcal{T}}_k(\boldsymbol{z}) = \kappa \boldsymbol{z} + \boldsymbol{T}_k \overline{\boldsymbol{z}}$, joka on tavalla (1.1) ilmaistuna kolmilävistäjämuotoa

$$\boldsymbol{\mathcal{T}}_k = \begin{bmatrix} \widehat{\boldsymbol{\alpha}}_1 & \widehat{\boldsymbol{\beta}}_2 & & & \\ \widehat{\boldsymbol{\beta}}_2 & \widehat{\boldsymbol{\alpha}}_2 & \widehat{\boldsymbol{\beta}}_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \widehat{\boldsymbol{\beta}}_{k-1} & \widehat{\boldsymbol{\alpha}}_{k-1} & \widehat{\boldsymbol{\beta}}_k \\ & & & \widehat{\boldsymbol{\beta}}_k & \widehat{\boldsymbol{\alpha}}_k \end{bmatrix},$$

missä skalaarioperaattorit ovat $\widehat{\boldsymbol{\alpha}}_j(z) = \kappa z + \alpha_j \bar{z}$, $\widehat{\boldsymbol{\beta}}_j(z) = \beta_j \bar{z}$. Oletetaan, että sillä on reaalilineaarinen LU-hajotelma, jonka on tällöin oltava muotoa

$$\boldsymbol{\mathcal{T}}_k = \boldsymbol{\mathcal{L}}_k \boldsymbol{\mathcal{U}}_k = \begin{bmatrix} \mathbf{1} & & & & \\ \boldsymbol{\lambda}_2 & \mathbf{1} & & & \\ & \boldsymbol{\lambda}_3 & \ddots & & \\ & & \ddots & \mathbf{1} & \\ & & & \boldsymbol{\lambda}_k & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 & \widehat{\boldsymbol{\beta}}_2 & & & \\ & \boldsymbol{\nu}_2 & \widehat{\boldsymbol{\beta}}_3 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{\nu}_{k-1} & \widehat{\boldsymbol{\beta}}_k \\ & & & & \boldsymbol{\nu}_k \end{bmatrix},$$

missä $\boldsymbol{\lambda}_k$ ja $\boldsymbol{\nu}_k$ syntyvät Gaussin eliminoinnissa ja niille saadaan kaavat

$$\begin{aligned}\boldsymbol{\nu}_1 &= \widehat{\boldsymbol{\alpha}}_1,\\ \boldsymbol{\lambda}_k &= \widehat{\boldsymbol{\beta}}_k \boldsymbol{\nu}_{k-1}^{-1},\\ \boldsymbol{\nu}_k &= \widehat{\boldsymbol{\alpha}}_k - \boldsymbol{\lambda}_k \widehat{\boldsymbol{\beta}}_k.\end{aligned}$$

Oletetaan $\boldsymbol{\mathcal{T}}_k$ vielä kääntyväksi, jolloin kaavasta (3.30) tulee

$$\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{\mathcal{Q}}_k \boldsymbol{\mathcal{U}}_k^{-1} \boldsymbol{\mathcal{L}}_k^{-1}(\|\boldsymbol{r}_0\|\boldsymbol{e}_1).$$

Merkitään

$$\boldsymbol{\mathcal{P}}_k = \boldsymbol{\mathcal{Q}}_k \boldsymbol{\mathcal{U}}_k^{-1}, \qquad \boldsymbol{y}_k = \boldsymbol{\mathcal{L}}_k^{-1}(\|\boldsymbol{r}_0\|\boldsymbol{e}_1), \tag{3.37}$$

jonka jälkeen likimääräisratkaisu saadaan kaavasta $\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{\mathcal{P}}_k(\boldsymbol{y}_k)$. Olkoot $\boldsymbol{\pi}_1$, $\ldots$, $\boldsymbol{\pi}_k$ operaattorin $\boldsymbol{\mathcal{P}}_k$ sarakkeet. Tällöin ensimmäisestä kaavasta (3.37) nähdään, että

$$\boldsymbol{\pi}_k = (\boldsymbol{q}_k - \boldsymbol{\pi}_{k-1}\widehat{\boldsymbol{\beta}}_k)\boldsymbol{\nu}_k^{-1}. \tag{3.38}$$

Merkitään η_k:lla vektorin $\boldsymbol{y}_k$ viimeistä komponenttia, jolloin toisesta kaavasta (3.37) saadaan

$$\eta_k = -\boldsymbol{\lambda}_k(\eta_{k-1}), \tag{3.39}$$

koska alakolmio-operaattoreille pätee

$$\boldsymbol{\mathcal{L}}_k = \begin{bmatrix} \mathbf{1} & & & & \\ \boldsymbol{\lambda}_2 & \mathbf{1} & & & \\ & \boldsymbol{\lambda}_3 & \ddots & & \\ & & \ddots & \mathbf{1} & \\ & & & \boldsymbol{\lambda}_{k-1} & \mathbf{1} \\ & & & & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & & & & \\ & \mathbf{1} & & & \\ & & \ddots & & \\ & & & \mathbf{1} & \\ & & & & \mathbf{1} \\ & & & & \boldsymbol{\lambda}_k \quad \mathbf{1} \end{bmatrix}.$$

Nyt likimääräisratkaisulle saadaan kaava

$$\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{\mathcal{P}}_k(\boldsymbol{y}_k) = \boldsymbol{z}_0 + \begin{bmatrix} \boldsymbol{\mathcal{P}}_{k-1} & \boldsymbol{\pi}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_{k-1} \\ \eta_k \end{bmatrix} = \boldsymbol{z}_{k-1} + \boldsymbol{\pi}_k(\eta_k). \tag{3.40}$$

Kun likimääräisratkaisu $\boldsymbol{z}_{k-1}$ on laskettu, se voidaan päivittää ratkaisuksi $\boldsymbol{z}_k$ laskemalla uusi kantavektori $\boldsymbol{q}_k$ yhdellä kierroksella algoritmia 3.7.1. Muistissa tarvitsee tällöin olla vain $\boldsymbol{q}_{k-1}$ ja $\boldsymbol{q}_{k-2}$. Sen jälkeen voidaan käyttää kaavaa (3.38) $\boldsymbol{\pi}_k$:n päivittämiseksi. Muistissa tarvitsee olla edellinen $\boldsymbol{\pi}_{k-1}$. Vielä kaavasta (3.39) saadaan päivitettyä η_k edellisen η_{k-1}:n perusteella. Lopuksi $\boldsymbol{z}_k$ lasketaan kaavasta (3.40). Tämä on kirjoitettu algoritmiksi 3.7.2.

Ratkaisun tarkkuutta on mahdollista arvioida kaavalla (3.33). Jäännökselle $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{\mathcal{M}}_\kappa(\boldsymbol{z}_k)$ pätee

$$\|\boldsymbol{r}_k\| = \beta_{k+1}|\boldsymbol{e}_k^T \boldsymbol{u}_k|,$$

missä $\boldsymbol{u}_k = \boldsymbol{\mathcal{U}}_k^{-1}\boldsymbol{\mathcal{L}}_k^{-1}(\|\boldsymbol{r}_0\|\boldsymbol{e}_1) = \boldsymbol{\mathcal{U}}_k^{-1}(\boldsymbol{y}_k)$. Täten

$$\|\boldsymbol{r}_k\| = \beta_{k+1}|\boldsymbol{\nu}_k^{-1}(\eta_k)| = |\boldsymbol{\lambda}_{k+1}(\eta_k)| = |\eta_{k+1}|. \tag{3.41}$$

Algoritmi 3.7.2 Symmetrisen $\boldsymbol{M}_{\#}$ ratkaisumenetelmä. K on suurin sallittu iteraatioiden lukumäärä.

1: $\boldsymbol{q}_1 \leftarrow \boldsymbol{r}_0/\|\boldsymbol{r}_0\|, \quad \beta_1 \leftarrow 0, \quad \boldsymbol{q}_0 \leftarrow 0, \quad \boldsymbol{p}_0 \leftarrow 0, \quad \boldsymbol{p}_{a,0} \leftarrow 0$
2: $\eta_1 \leftarrow \|\boldsymbol{r}_0\|, \quad \lambda_1 \leftarrow 0, \quad \lambda_{a,1} \leftarrow 0$
3: **for** $k = 1$ to K **do**
4: $\quad \boldsymbol{w} \leftarrow \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_k - \beta_k \boldsymbol{q}_{k-1}$
5: $\quad \alpha_k \leftarrow \langle \boldsymbol{w}, \boldsymbol{q}_k \rangle$
6: $\quad \begin{bmatrix} u_k & u_{a,k} \end{bmatrix} \leftarrow \begin{bmatrix} \kappa & \alpha_k \end{bmatrix} - \beta_k \begin{bmatrix} \lambda_{a,k} & \lambda_k \end{bmatrix}$
7: $\quad \begin{bmatrix} \boldsymbol{p}_k & \boldsymbol{p}_{a,k} \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{q}_k - \beta_k \boldsymbol{p}_{a,k-1} & -\beta_k \boldsymbol{p}_{k-1} \end{bmatrix} \begin{bmatrix} u_k & u_{a,k} \\ \overline{u}_{a,k} & \overline{u}_k \end{bmatrix}^{-1}$
8: $\quad \boldsymbol{z}_k \leftarrow \boldsymbol{z}_{k-1} + \eta_k \boldsymbol{p}_k + \overline{\eta}_k \boldsymbol{p}_{a,k}$
9: $\quad$ Jos $\boldsymbol{z}_k$ on riittävän tarkka, lopetetaan algoritmi.
10: $\quad \boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_k \boldsymbol{q}_k$
11: $\quad \beta_{k+1} \leftarrow \|\boldsymbol{w}\|$
12: $\quad \boldsymbol{q}_{k+1} \leftarrow \boldsymbol{w}/\beta_{k+1}$
13: $\quad \begin{bmatrix} \lambda_{k+1} & \lambda_{a,k+1} \end{bmatrix} \leftarrow \begin{bmatrix} 0 & \beta_{k+1} \end{bmatrix} \begin{bmatrix} u_k & u_{a,k} \\ \overline{u}_{a,k} & \overline{u}_k \end{bmatrix}^{-1}$
14: $\quad \eta_{k+1} \leftarrow -\lambda_{k+1}\eta_k - \lambda_{a,k+1}\overline{\eta}_k$
15: **end for**

3.7.2 Menetelmä vinosymmetriselle $\boldsymbol{M}_{\#}$

Edellä oleva symmetrisen $\boldsymbol{M}_{\#}$ matriisin menetelmä saattaa rikkoutua, koska oletettua LU-hajotelmaa ei ole. Vinosymmetrisillä matriiseilla ei tätä ongelmaa esiinny, kunhan $\kappa \neq 0$. Seuraavassa asetetaankin $\kappa = 1$ ja ratkaistavana on siten yhtälö $\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, missä $\boldsymbol{M}_{\#}$ on vinosymmetrinen. Tuloksena on matriisien liittogradienttimenetelmää muistuttava menetelmä.

Yhtälöstä (3.28) saadaan nyt

$$\boldsymbol{H}_k^T = (\boldsymbol{Q}_k^* \boldsymbol{M}_{\#} \overline{\boldsymbol{Q}}_k)^T = \overline{\boldsymbol{Q}}_k^T \boldsymbol{M}_{\#}{}^T (\boldsymbol{Q}_k^*)^T = \boldsymbol{Q}_k^* (-\boldsymbol{M}_{\#}) \overline{\boldsymbol{Q}}_k = -\boldsymbol{H}_k,$$

joten Hessenbergin matriisi $\boldsymbol{H}_k$ on myös vinosymmetrinen. Täten se on muotoa

$$\boldsymbol{T}_k = \begin{bmatrix} 0 & -\beta_2 & & & \\ \beta_2 & 0 & -\beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & 0 & -\beta_k \\ & & & \beta_k & 0 \end{bmatrix}.$$

Tällöin $\beta_j = h_{j,j-1}$ ja Arnoldin menetelmästä (3.23)-(3.25) tulee vielä edellistä kohtaakin yksinkertaisempi, koska $\langle \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}, \boldsymbol{q} \rangle = 0$ kaikilla vektoreilla $\boldsymbol{q}$. Asetetaan $\beta_1 = 0$, $\boldsymbol{q}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$ ja lasketaan vektorit $\boldsymbol{q}_2, \boldsymbol{q}_3, \ldots$ iteroimalla seuraavia kaavoja aloittaen arvosta $j = 1$.

$$\boldsymbol{w}_j = \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_j + \beta_j \boldsymbol{q}_{j-1}, \tag{3.42}$$

$$\beta_{j+1} = \|\boldsymbol{w}_j\|. \tag{3.43}$$

Menetelmä pysäytetään, jos $\beta_{j+1} = 0$. Muutoin asetetaan $\boldsymbol{q}_{j+1} = \boldsymbol{w}_j/\beta_{j+1}$. Vaikka tarkassa aritmetiikassa ortonormalisointia ei enää tarvitakaan, niin numeerisessa

laskennassa tästä saattaa olla hyötyä. Tätä varten voidaan vaihtaa (3.42) tilalle kaava

$$\widetilde{\boldsymbol{w}}_j = \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_j + \beta_j \boldsymbol{q}_{j-1}, \quad \boldsymbol{w}_j = \widetilde{\boldsymbol{w}}_j - \langle \widetilde{\boldsymbol{w}}_j, \boldsymbol{q}_j \rangle \, \boldsymbol{q}_j. \tag{3.44}$$

Merkitään $\boldsymbol{\mathcal{T}}_k(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{T}_k\overline{\boldsymbol{z}}$, joka on kolmilävistäjämuotoa

$$\boldsymbol{\mathcal{T}}_k = \begin{bmatrix} \mathbf{1} & -\widehat{\boldsymbol{\beta}}_2 & & & \\ \widehat{\boldsymbol{\beta}}_2 & \mathbf{1} & -\widehat{\boldsymbol{\beta}}_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \widehat{\boldsymbol{\beta}}_{k-1} & \mathbf{1} & -\widehat{\boldsymbol{\beta}}_k \\ & & & \widehat{\boldsymbol{\beta}}_k & \mathbf{1} \end{bmatrix},$$

missä skalaarioperaattorit ovat $\widehat{\boldsymbol{\beta}}_j(z) = \beta_j \overline{z}$. Sen LU-hajotelman (mikäli olemassa) on oltava muotoa

$$\boldsymbol{\mathcal{T}}_k = \boldsymbol{\mathcal{L}}_k\boldsymbol{\mathcal{U}}_k = \begin{bmatrix} \mathbf{1} & & & \\ \boldsymbol{\lambda}_2 & \mathbf{1} & & \\ & \boldsymbol{\lambda}_3 & \ddots & \\ & & \ddots & \mathbf{1} \\ & & & \boldsymbol{\lambda}_k \quad \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 & -\widehat{\boldsymbol{\beta}}_2 & & & \\ & \boldsymbol{\nu}_2 & -\widehat{\boldsymbol{\beta}}_3 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{\nu}_{k-1} & -\widehat{\boldsymbol{\beta}}_k \\ & & & & \boldsymbol{\nu}_k \end{bmatrix},$$

missä $\boldsymbol{\lambda}_k$ ja $\boldsymbol{\nu}_k$ syntyvät Gaussin eliminoinnissa, mistä niille saadaan kaavat

$$\begin{aligned} \boldsymbol{\nu}_1 &= \mathbf{1}, \\ \boldsymbol{\lambda}_k &= \widehat{\boldsymbol{\beta}}_k \boldsymbol{\nu}_{k-1}^{-1}, \\ \boldsymbol{\nu}_k &= \mathbf{1} + \boldsymbol{\lambda}_k \widehat{\boldsymbol{\beta}}_k. \end{aligned}$$

Nähdään helposti, että nämä saadaan muotoon

$$\begin{aligned} \gamma_1 &= 1, \\ l_k &= \beta_k/\gamma_{k-1}, \\ \gamma_k &= 1 + l_k\beta_k, \end{aligned}$$

missä $\boldsymbol{\nu}_k(z) = \gamma_k z$ ja $\boldsymbol{\lambda}_k(z) = l_k\overline{z}$. Kaikilla j on $\gamma_j \geq 1$, joten LU-hajotelma on aina olemassa ja $\boldsymbol{\mathcal{T}}_k$ on kääntyvä. Kaavasta (3.30) saadaan

$$\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{Q}_k\boldsymbol{\mathcal{U}}_k^{-1}\boldsymbol{\mathcal{L}}_k^{-1}(\|\boldsymbol{r}_0\|\boldsymbol{e}_1)$$

ja kaavoista (3.38)-(3.40) tulee

$$\begin{aligned} \boldsymbol{\pi}_k &= (\boldsymbol{q}_k - \boldsymbol{\pi}_{k-1}\widehat{\boldsymbol{\beta}}_k)\boldsymbol{\nu}_k^{-1} = (1/\gamma_k)(\boldsymbol{q}_k - \boldsymbol{\pi}_{k-1}\widehat{\boldsymbol{\beta}}_k), \\ \eta_k &= -\boldsymbol{\lambda}_k(\eta_{k-1}) = -l_k\eta_{k-1}, \\ \boldsymbol{z}_k &= \boldsymbol{z}_{k-1} + \boldsymbol{\pi}_k(\eta_k), \end{aligned}$$

missä on käytetty huomiota, että nyt $\eta_j \in \mathbb{R}$ kaikilla j. Kaavoja saadaan vielä yksinkertaistettua merkitsemällä $\boldsymbol{\pi}_k(z) = \widetilde{\boldsymbol{p}}_k z + \widetilde{\boldsymbol{p}}_{\#,k}\overline{z}$ ja $\boldsymbol{p}_k = \widetilde{\boldsymbol{p}}_k + \widetilde{\boldsymbol{p}}_{\#,k}$. Koska $\eta_k \in \mathbb{R}$ ja $\widehat{\boldsymbol{\beta}}_k$ on anti-lineaarinen, niin kaavat saadaan muotoon

$$\boldsymbol{p}_k = \frac{1}{\gamma_k}(\boldsymbol{q}_k - \beta_k\boldsymbol{p}_{k-1}), \tag{3.45}$$

$$\eta_k = -l_k \eta_{k-1}, \tag{3.46}$$

$$\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \eta_k \boldsymbol{p}_k. \tag{3.47}$$

Näin saadaan algoritmi 3.7.3. Ratkaisun jäännöksen normille (3.41) pätee edelleen

$$\|\boldsymbol{r}_k\| = |\eta_{k+1}|.$$

Kunkin kierroksen laskuja varten täytyy muistissa pitää vain kaksi edellistä $\boldsymbol{q}$-vektoria ja yksi edellinen $\boldsymbol{p}$-vektori.

Algoritmi 3.7.3 Vinosymmetrisen $\boldsymbol{M}_\#$ ratkaisumenetelmä ($\kappa = 1$). K on suurin sallittu iteraatioiden lukumäärä.

1: $\boldsymbol{q}_1 \leftarrow \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$, $\beta_1 \leftarrow 0$, $\boldsymbol{q}_0 \leftarrow 0$, $\boldsymbol{p}_0 \leftarrow 0$, $\eta_1 \leftarrow \|\boldsymbol{r}_0\|$, $\gamma_1 \leftarrow 1$
2: **for** $k = 1$ to K **do**
3: $\quad \boldsymbol{p}_k \leftarrow \frac{1}{\gamma_k}(\boldsymbol{q}_k - \beta_k \boldsymbol{p}_{k-1})$
4: $\quad \boldsymbol{z}_k \leftarrow \boldsymbol{z}_{k-1} + \eta_k \boldsymbol{p}_k$
5: $\quad$ Jos $\boldsymbol{z}_k$ on riittävän tarkka, lopetetaan algoritmi.
6: $\quad \boldsymbol{w} \leftarrow \boldsymbol{M}_\# \overline{\boldsymbol{q}}_k + \beta_k \boldsymbol{q}_{k-1}$
7: $\quad \boldsymbol{w} \leftarrow \boldsymbol{w} - \langle \boldsymbol{w}, \boldsymbol{q}_k \rangle \boldsymbol{q}_k$
8: $\quad \beta_{k+1} \leftarrow \|\boldsymbol{w}\|$
9: $\quad \boldsymbol{q}_{k+1} \leftarrow \boldsymbol{w}/\beta_{k+1}$
10: $\quad \lambda_{k+1} \leftarrow \beta_{k+1}/\gamma_k$, $\gamma_{k+1} \leftarrow 1 + \lambda_{k+1}\beta_{k+1}$
11: $\quad \eta_{k+1} \leftarrow -\lambda_{k+1}\eta_k$
12: **end for**

3.8 $\mathbb{R}$-lineaarinen GMRES operaattorille $\mathcal{M}_\kappa$

Tämän kohdan menetelmä toimii yleisille matriiseille $\boldsymbol{M}_\#$. Likimääräisratkaisua yhtälölle $\mathcal{M}_\kappa(\boldsymbol{z}) = \boldsymbol{b}$ etsitään jälleen muodossa $\boldsymbol{z}_k \in \boldsymbol{z}_0 + \mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0)$, mutta nyt vaaditaan jäännöksen $\boldsymbol{r}_k = \boldsymbol{b} - \mathcal{M}_\kappa(\boldsymbol{z}_k)$ normi pienimmäksi mahdolliseksi. Tämä on yhtäpitävää ehdon $\boldsymbol{r}_k \perp \mathcal{M}_\kappa(\mathcal{K}_k(\mathcal{M}_\kappa, \boldsymbol{r}_0))$ kanssa. Kun matriisin $\boldsymbol{Q}_k$ sarakkeet on laskettu Arnoldin menetelmällä, normille saadaan yhtälön (3.26) mukaan

$$\begin{aligned}\|\boldsymbol{r}_k\| &= \|\boldsymbol{r}_0 - \kappa \boldsymbol{Q}_k \boldsymbol{u}_k - \boldsymbol{M}_\# \overline{\boldsymbol{Q}}_k \overline{\boldsymbol{u}_k}\| = \|\boldsymbol{r}_0 - \kappa \boldsymbol{Q}_k \boldsymbol{u}_k - \boldsymbol{Q}_{k+1} \widehat{\boldsymbol{H}}_k \overline{\boldsymbol{u}_k}\| \\ &= \left\|\boldsymbol{Q}_{k+1}(\|\boldsymbol{r}_0\| \widehat{\boldsymbol{e}}_1 - \kappa \widehat{\boldsymbol{I}}_k \boldsymbol{u}_k - \widehat{\boldsymbol{H}}_k \overline{\boldsymbol{u}_k})\right\| = \left\|\|\boldsymbol{r}_0\| \widehat{\boldsymbol{e}}_1 - \kappa \widehat{\boldsymbol{I}}_k \boldsymbol{u}_k - \widehat{\boldsymbol{H}}_k \overline{\boldsymbol{u}_k}\right\|, \end{aligned} \tag{3.48}$$

missä $\widehat{\boldsymbol{e}}_1$ on $\mathbb{C}^{k+1}$:n ensimmäinen luonnollinen kantavektori ja $\widehat{\boldsymbol{I}}_k$ on yksikkömatriisi $\boldsymbol{I}_k$ lisättynä yhdellä nollarivillä. Tämä pienimmän neliösumman tehtävä voidaan ratkaista laskemalla QR-hajotelma operaattorille $\widehat{\mathcal{H}}_k : \mathbb{C}^k \to \mathbb{C}^{k+1}$, $\widehat{\mathcal{H}}_k(\boldsymbol{u}) = \kappa \widehat{\boldsymbol{I}}_k \boldsymbol{u} + \widehat{\boldsymbol{H}}_k \overline{\boldsymbol{u}}$. Käyttäen skalaarioperaattorimuotoa (1.1), voidaan kirjoittaa

$$\widehat{\mathcal{H}}_k = \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1k} \\ \mu_{21} & \mu_{22} & \mu_{23} & & \mu_{2k} \\ & \mu_{32} & \mu_{33} & & \mu_{3k} \\ & & \ddots & & \vdots \\ & & & \mu_{k,k-1} & \mu_{k,k} \\ & & & & \mu_{k+1,k} \end{bmatrix}, \tag{3.49}$$

missä skalaarioperaattorit ovat

$$\boldsymbol{\mu}_{ij}(z) = \begin{cases} h_{ij}\overline{z}, & \text{kun } i \neq j, \\ \kappa z + h_{ij}\overline{z}, & \text{kun } i = j. \end{cases}$$

Matriisien GMRES-menetelmässä kohdassa 3.4 käytettiin Givensin rotaatioita Hessenbergin matriisin muuntamiseksi yläkolmiomuotoon. Nyt tarvitaan vastaavanlainen reaalilineaarinen operaatio. Myöhemmin nähdään, että mille tahansa skalaarioperaattoreille $\boldsymbol{\mu}$ ja $\boldsymbol{\nu}$ on olemassa unitaarinen reaalilineaarinen operaattori $\boldsymbol{\mathcal{U}} : \mathbb{C}^2 \to \mathbb{C}^2$ siten, että

$$\begin{bmatrix} \boldsymbol{g}_{11} & \boldsymbol{g}_{12} \\ \boldsymbol{g}_{21} & \boldsymbol{g}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\nu} \end{bmatrix} = \begin{bmatrix} \# \\ 0 \end{bmatrix}, \tag{3.50}$$

missä skalaarioperaattorit $\boldsymbol{g}_{ij}$ muodostavat $\boldsymbol{\mathcal{U}}$:n. QR-hajotelman lausetta 1.2.4 vastaten, voidaan $\widehat{\boldsymbol{\mathcal{H}}}_k$ muuntaa yläkolmiomuotoon kertomalla $\mathbb{C}^{k+1} \to \mathbb{C}^{k+1}$ operaattoreilla muotoa

$$\boldsymbol{\mathcal{G}}_i = \begin{bmatrix} \mathbf{1} & & & & & & & \\ & \ddots & & & & & & \\ & & \mathbf{1} & & & & & \\ & & & \boldsymbol{g}_{11}^{(i)} & \boldsymbol{g}_{12}^{(i)} & & & \\ & & & \boldsymbol{g}_{21}^{(i)} & \boldsymbol{g}_{22}^{(i)} & & & \\ & & & & & \mathbf{1} & & \\ & & & & & & \ddots & \\ & & & & & & & \mathbf{1} \end{bmatrix}. \tag{3.51}$$

Merkitään $\widehat{\boldsymbol{\mathcal{G}}}_k = \boldsymbol{\mathcal{G}}_k \cdots \boldsymbol{\mathcal{G}}_2 \boldsymbol{\mathcal{G}}_1$, $\widehat{\boldsymbol{\mathcal{R}}}_k = \widehat{\boldsymbol{\mathcal{G}}}_k \widehat{\boldsymbol{\mathcal{H}}}_k$ ja $\widehat{\boldsymbol{\eta}}_k = \widehat{\boldsymbol{\mathcal{G}}}_k(\widehat{e}_1)$. Olkoon vektori $\boldsymbol{\eta}_k$ tämän k ensimmäistä komponenttia ja ζ_k viimeinen komponentti. Merkitään vielä $\boldsymbol{\mathcal{R}}_k$:lla sitä $\mathbb{C}^k \to \mathbb{C}^k$-operaattoria, joka saadaan poistamalla $\widehat{\boldsymbol{\mathcal{R}}}_k$:sta viimeinen (nolla-)skalaarioperaattoririvi.

Käyttäen operaattoreiden $\boldsymbol{\mathcal{G}}_i$ unitaarisuutta, tulee (3.48) muotoon

$$\begin{aligned} \|\boldsymbol{r}_k\|^2 &= \left\| \|\boldsymbol{r}_0\| \widehat{e}_1 - \widehat{\boldsymbol{\mathcal{H}}}_k(\boldsymbol{u}_k) \right\|^2 = \left\| \widehat{\boldsymbol{\mathcal{G}}}_k \left(\|\boldsymbol{r}_0\| \widehat{e}_1 - \widehat{\boldsymbol{\mathcal{H}}}_k(\boldsymbol{u}_k) \right) \right\|^2 \\ &= \left\| \|\boldsymbol{r}_0\| \widehat{\boldsymbol{\eta}}_k - \widehat{\boldsymbol{\mathcal{R}}}_k(\boldsymbol{u}_k) \right\|^2 = \left\| \|\boldsymbol{r}_0\| \boldsymbol{\eta}_k - \boldsymbol{\mathcal{R}}_k(\boldsymbol{u}_k) \right\|^2 + \|\boldsymbol{r}_0\|^2 |\zeta_k|^2. \end{aligned} \tag{3.52}$$

Operaattorin $\widehat{\boldsymbol{\mathcal{H}}}_k$ $2(k+1) \times 2k$-reaalimatriisiesityksen rangi on $2k$ kohdan (3.25) perusteella. Näin myös $\widehat{\boldsymbol{\mathcal{R}}}_k$:n ja $\boldsymbol{\mathcal{R}}_k$:n reaalimatriisien rangi on $2k$, joten $\boldsymbol{\mathcal{R}}_k$ on kääntyvä. Ratkaistaan yläkolmioyhtälö

$$\boldsymbol{\mathcal{R}}_k(\boldsymbol{u}_k) = \|\boldsymbol{r}_0\| \boldsymbol{\eta}_k$$

taaksepäin sijoituksin. Lävistäjällä olevien skalaarioperaattoreiden täytyy olla kääntyviä, kun $\boldsymbol{\mathcal{R}}_k$ on kääntyvä. Tämän jälkeen GMRES-menetelmän likimääräisratkaisu lasketaan kaavasta $\boldsymbol{z}_k = \boldsymbol{z}_0 + \boldsymbol{Q}_k \boldsymbol{u}_k$. Tällöin $\|\boldsymbol{r}_k\| = \|\boldsymbol{r}_0\| |\zeta_k|$.

Arnoldin menetelmän murtuessa $h_{k+1,k} = 0$ $(\boldsymbol{w}_k = 0)$, jolloin GMRES antaa tarkan ratkaisun alkuperäiselle yhtälölle. Tällöin $\boldsymbol{\mathcal{G}}_k$ voidaan valita yksikköoperaattoriksi, joten $\zeta_k = 0$ ja kaavasta (3.52) nähdään $\boldsymbol{r}_k = 0$.

Esitetty menetelmä on kirjoitettu algoritmiksi 3.8.1. Huomaa, että jäännöksen normi $\|\boldsymbol{r}_k\|$ on algoritmin jokaisella kierroksella $\beta |(\boldsymbol{v}_j)_{j+1}|$ ja sitä käyttäen algoritmi voitaisiin muuttaa lopettamaan iterointi heti, kun riittävä tarkkuus on saavutettu.

Unitaarisina operaattoreina (3.50) voidaan käyttää kohdassa 1.2 esiteltyjä Householderin muunnoksia. Toinen mahdollisuus on käyttää seuraavan lauseen tarjoamia Givensin rotaatioiden tapaisia reaalilineaarisia operaattoreita. Huomaa, että kaavan (3.25) mukaan $h_{j+1,j}$ on reaaliluku.

Lause 3.8.1. *Olkoot skalaarioperaattorit*

$$\boldsymbol{\mu}(z) = \kappa z + a\overline{z}, \qquad \boldsymbol{\nu}(z) = b\overline{z},$$

missä $\kappa, a \in \mathbb{C}$ *ja* $b \in \mathbb{R}$, $b \neq 0$. *Asetetaan*

$$\widehat{\boldsymbol{U}} = \begin{bmatrix} \overline{\kappa} & 0 \\ 0 & \overline{\kappa} \end{bmatrix}, \qquad \widehat{\boldsymbol{U}}_{\#} = \begin{bmatrix} a & b \\ -b & \overline{a} \end{bmatrix}, \qquad \widehat{\boldsymbol{u}}(\boldsymbol{z}) = \widehat{\boldsymbol{U}}\boldsymbol{z} + \widehat{\boldsymbol{U}}_{\#}\overline{\boldsymbol{z}},$$

$$x = |\kappa|^2 + |a|^2 + b^2, \qquad y = 2a\kappa,$$

$$r_1 = r_2 = \sqrt{\frac{x + \sqrt{x^2 - |y|^2}}{2}}, \qquad s_1 = \frac{a\kappa}{r_1}, \qquad s_2 = \frac{\overline{a}\kappa}{r_2},$$

$$\boldsymbol{\beta}_i(z) = r_i z + s_i \overline{z}.$$

Tällöin operaattori $\boldsymbol{\mathcal{U}} : \mathbb{C}^2 \to \mathbb{C}^2$,

$$\boldsymbol{\mathcal{U}} = \widehat{\boldsymbol{\mathcal{U}}} \circ \begin{bmatrix} \boldsymbol{\beta}_1^{-1} & \\ & \boldsymbol{\beta}_2^{-1} \end{bmatrix}$$

on unitaarinen ja

$$\boldsymbol{\mathcal{U}} \circ \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\nu} \end{bmatrix} = \begin{bmatrix} \# \\ 0 \end{bmatrix}.$$

Todistus. Suoraan laskemalla nähdään

$$\widehat{\boldsymbol{\mathcal{U}}}^* \widehat{\boldsymbol{\mathcal{U}}} = \begin{bmatrix} \boldsymbol{\beta}_1^2 & \\ & \boldsymbol{\beta}_2^2 \end{bmatrix},$$

joten

$$\boldsymbol{\mathcal{U}}^* \boldsymbol{\mathcal{U}} = \begin{bmatrix} \boldsymbol{\beta}_1^{-1} & \\ & \boldsymbol{\beta}_2^{-1} \end{bmatrix} \widehat{\boldsymbol{\mathcal{U}}}^* \widehat{\boldsymbol{\mathcal{U}}} \begin{bmatrix} \boldsymbol{\beta}_1^{-1} & \\ & \boldsymbol{\beta}_2^{-1} \end{bmatrix} = \boldsymbol{I}$$

ja täten $\boldsymbol{\mathcal{U}}$ on unitaarinen. Niin ikään suoralla laskulla todetaan

$$\widehat{\boldsymbol{\mathcal{U}}} \circ \begin{bmatrix} \boldsymbol{\beta}_1^{-1} \circ \boldsymbol{\mu} \\ \boldsymbol{\beta}_2^{-1} \circ \boldsymbol{\nu} \end{bmatrix} = \begin{bmatrix} \# \\ 0 \end{bmatrix}.$$

□

Esimerkki 1.1.5 sisältää kaavan edellisessä lauseessa olevien $\boldsymbol{\beta}_i^{-1}$ laskemiseksi.

Yhtälö $\boldsymbol{\mathcal{M}}_\kappa(\boldsymbol{z}) = \boldsymbol{b}$ voitaisiin myös ratkaista kirjoittamalla se kaksi kertaa suuremmaksi reaaliseksi yhtälöryhmäksi. Käyttäen kohdan (1.3) esitystä saadaan ($\kappa = c + id$)

$$\boldsymbol{A} = \begin{bmatrix} c\boldsymbol{I} + \mathrm{Re}(\boldsymbol{M}_\#) & -d\boldsymbol{I} + \mathrm{Im}(\boldsymbol{M}_\#) \\ d\boldsymbol{I} + \mathrm{Im}(\boldsymbol{M}_\#) & c\boldsymbol{I} - \mathrm{Re}(\boldsymbol{M}_\#) \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \mathrm{Re}(\boldsymbol{z}) \\ \mathrm{Im}(\boldsymbol{z}) \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} \mathrm{Re}(\boldsymbol{b}) \\ \mathrm{Im}(\boldsymbol{b}) \end{bmatrix},$$

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}.$$

Algoritmi 3.8.1 GMRES-menetelmä yhtälön $\mathcal{M}_\kappa(z) = b$ likimääräiseksi ratkaisemiseksi. Suorittaa k iteraatiota tai pysähtyy, jos tarkka ratkaisu saavutetaan. Alkuarvaus on z_0.

$r_0 \leftarrow b - \mathcal{M}_\kappa(z_0)$, $\beta \leftarrow \|r_0\|$, $q_1 \leftarrow r_0/\beta$, $v \leftarrow 1$, $\widehat{\mathcal{R}} \leftarrow [\,]$
for $j = 1$ to k **do**
  $w \leftarrow M_\# \overline{q}_j$
  for $i = 1$ to j **do**
    $h_{ij} \leftarrow \langle w, q_i \rangle$
    $w \leftarrow w - h_{ij} q_i$
  end for
  $h_{j+1,j} \leftarrow \|w\|$
  $s \leftarrow \begin{bmatrix} h_{1j} & h_{2j} & \cdots & h_{j+1,j} \end{bmatrix}^T$
  Olkoon $\mathcal{S} : \mathbb{C} \to \mathbb{C}^{j+1}$ siten, että $\mathcal{S}(z) = \kappa e_j z + s\overline{z}$
  for $i = 1$ to $j - 1$ **do**
    $\mathcal{S}_{i:(i+1)} \leftarrow \mathcal{U}_i \circ \mathcal{S}_{i:(i+1)}$
  end for
  Valitaan unitaari $\mathcal{U}_j : \mathbb{C}^2 \to \mathbb{C}^2$ siten, että $\mathcal{U}_j \circ \mathcal{S}_{j:(j+1)} = \begin{bmatrix} \# & 0 \end{bmatrix}^T$
  $\mathcal{S}_{j:(j+1)} \leftarrow \mathcal{U}_j \circ \mathcal{S}_{j:(j+1)}$
  $\widehat{\mathcal{R}} \leftarrow \begin{bmatrix} \widehat{\mathcal{R}} \\ \mathbf{0} \end{bmatrix}$, $\widehat{\mathcal{R}} \leftarrow \begin{bmatrix} \widehat{\mathcal{R}} & \mathcal{S} \end{bmatrix}$, missä $\mathbf{0}$ on $1 \times (j-1)$-nollaoperaattori
  $v \leftarrow \begin{bmatrix} v \\ 0 \end{bmatrix} \in \mathbb{C}^{j+1}$, $v_{j:(j+1)} \leftarrow \mathcal{U}_j(v_{j:(j+1)})$
  if $h_{j+1,j} = 0$ **then**
    break
  end if
  $q_{j+1} \leftarrow w/h_{j+1,j}$
end for
$Q \leftarrow [q_1 \cdots q_j]$, $\mathcal{R} \leftarrow \widehat{\mathcal{R}}_{1:j,*}$
Ratkaistaan u yhtälöstä $\mathcal{R}(u) = \beta v_{1:j}$
$z \leftarrow z_0 + Qu$, jäännöksen normi on $\beta|v_{j+1}|$

Kun matriisien GMRES etsii jäännöksen minimoivan ratkaisun Krylovin aliavaruudesta ($v = f - Ax_0$)

$$\mathcal{K}_k(A, v) = \operatorname{span}\left\{v, Av, A^2v, \ldots, A^{k-1}v\right\},$$

niin reaalilineaarinen GMRES etsii ratkaisun avaruudesta ($r_0 = b - \mathcal{M}_\kappa(z_0)$)

$$\mathcal{K}_k(\mathcal{M}_\kappa, r_0) = \operatorname{span}\left\{r_0, \mathcal{M}_0(r_0), \mathcal{M}_0^2(r_0), \ldots, \mathcal{M}_0^{k-1}(r_0)\right\}.$$

Edellinen on $\mathbb{R}$-kertoiminen ja jälkimmäinen on $\mathbb{C}$-kertoiminen, joten samaistuksen $z \leftrightarrow \begin{bmatrix} \mathrm{Re}(z) \\ \mathrm{Im}(z) \end{bmatrix}$ mielessä on aidosti $\mathcal{K}_k(A, v) \subset \mathcal{K}_k(\mathcal{M}_\kappa, r_0)$ ja siten reaalilineaarinen GMRES etsii kullakin iteraatiokierroksella jäännöksen minimoivan ratkaisun laajemmasta joukosta. Täten se suppenee aina vähintään yhtä nopeasti kuin matriisien GMRES. Numeeristen kokeiden mukaan se on yleensä nopeampi (ks. kohta 5.1).

3.8.1 MINRES symmetriselle $M_\#$

Kohdan 3.7.1 menetelmän ongelmana on sen mahdollinen rikkoutuminen LU-hajotelman puuttumisen vuoksi. Edellisen kohdan GMRES voidaan erikoistaa symmetrisille $M_\#$ ilman tällaista ongelmaa ja silti saavuttaa kohtaa 3.7.1 vastaava säästö kantavektoreiden suhteen.

Kohdassa 3.7.1 kuvatulla Lanczosin menetelmällä edellisen kohdan GMRESin Hessenbergin operaattori (3.49) tulee kolmilävistäjämuotoon

$$\widehat{\mathcal{T}}_k = \begin{bmatrix} \widehat{\alpha}_1 & \widehat{\beta}_2 & & & \\ \widehat{\beta}_2 & \widehat{\alpha}_2 & \widehat{\beta}_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \widehat{\beta}_{k-1} & \widehat{\alpha}_{k-1} & \widehat{\beta}_k \\ & & & \widehat{\beta}_k & \widehat{\alpha}_k \\ & & & & \widehat{\beta}_{k+1} \end{bmatrix},$$

missä skalaarioperaattorit ovat $\widehat{\alpha}_j(z) = \kappa z + \alpha_j \bar{z}$, $\widehat{\beta}_j(z) = \beta_j \bar{z}$. Kun tämä muunnetaan yläkolmiomuotoon edellisen kohdan muotoisilla unitaarioperaattoreilla (3.51), saadaan kolmilävistäjinen yläkolmio-operaattori $\widehat{\mathcal{R}}_k$. Merkitään edelleen $\mathcal{R}_k = (\rho_{ij})$ sitä operaattoria, joka saadaan poistamalla $\widehat{\mathcal{R}}_k$:n viimeinen rivi. Määritellään sitten operaattori $\mathcal{P}_k = \mathcal{Q}_k \mathcal{R}_k^{-1}$ ja olkoot $\pi_1, \ldots, \pi_k$ sen sarakkeet. Koska $\mathcal{P}_k \mathcal{R}_k = \mathcal{Q}_k$, niin

$$\pi_k = (q_k - \pi_{k-1}\rho_{k-1,k} - \pi_{k-2}\rho_{k-2,k})\rho_{k,k}^{-1}.$$

Näin ollen, muistamalla vain kaksi edellistä π_{k-2} ja π_{k-1}, saadaan laskettua seuraava π_k. Likimääräisratkaisua voidaan päivittää kullakin kierroksella kaavalla

$$\begin{aligned} z_k &= z_0 + \mathcal{Q}_k \mathcal{R}_k^{-1}(\|r_0\|\eta_k) = z_0 + \begin{bmatrix} \mathcal{P}_{k-1} & \pi_k \end{bmatrix} (\|r_0\|\eta_k) \\ &= z_{k-1} + \|r_0\|\pi_k((\eta_k)_k). \end{aligned}$$

Edellä kuvatusta menetelmästä saadaan algoritmi 3.8.2. Jäännöksen normi $\|r_k\|$ olisi nytkin helposti käytettävissä jokaisella kierroksella laskemalla $\beta|(v_j)_{j+1}|$.

3.9 C-linearisoitu yhtälö

Kerrotaan yhtälö $\mathcal{M}_0(z) = b$ vasemmalta puolittain operaattorilla $\mathcal{M}_0^{k-1}$, jolloin parillisilla k saadaan C-lineaarinen (matriisi-)yhtälö $(M_\#\overline{M_\#})^{k/2} z = \widehat{b}$, missä $\widehat{b} = (M_\#\overline{M_\#})^{\frac{k}{2}-1} M_\# \bar{b}$. Kun $\kappa \neq 0$ ja tarkastellaan yhtälöä $\mathcal{M}_\kappa(z) = b$, voidaan kertoa vasemmalta polynomilla

$$p(\mathcal{M}_0) = \sum_{j=0}^{k-1} (-\frac{1}{\kappa}\mathcal{M}_0)^j \frac{1}{\kappa},$$

mikä saadaan katkaisemalla käänteisoperaattorin $(\kappa I + \mathcal{M}_0)^{-1} = \big(\kappa(I + \frac{1}{\kappa}\mathcal{M}_0)\big)^{-1} = (I + \frac{1}{\kappa}\mathcal{M}_0)^{-1}\frac{1}{\kappa}$ geometrinen sarja. Saadaan matriisiyhtälö

$$\big(I - \tfrac{1}{\kappa}(M_\#\overline{M_\#})^{k/2}\big) z = \widehat{b}, \qquad \widehat{b} = p(\mathcal{M}_0)(b). \tag{3.53}$$

Algoritmi 3.8.2 MINRES menetelmä symmetrisille $\boldsymbol{M}_{\#}$. Suorittaa k iteraatiota tai pysähtyy, jos tarkka ratkaisu saavutetaan. Alkuarvaus on $\boldsymbol{z}_0$.

$\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{\mathcal{M}}_\kappa(\boldsymbol{z}_0)$, $\beta \leftarrow \|\boldsymbol{r}_0\|$, $\boldsymbol{q}_1 \leftarrow \boldsymbol{r}_0/\beta$, $v_0 \leftarrow 1$
$\beta_1 \leftarrow 0$, $\boldsymbol{q}_0 \leftarrow 0$, $\boldsymbol{\pi}_0 \leftarrow 0$, $\boldsymbol{\pi}_{-1} \leftarrow 0$, $\boldsymbol{z} \leftarrow \boldsymbol{z}_0$
for $j = 1$ to k **do**
  $\boldsymbol{w} \leftarrow \boldsymbol{M}_{\#}\overline{\boldsymbol{q}}_j - \beta_j \boldsymbol{q}_{j-1}$
  $\alpha_j \leftarrow \langle \boldsymbol{w}, \boldsymbol{q}_j \rangle$
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_j \boldsymbol{q}_j$
  $\beta_{j+1} \leftarrow \|\boldsymbol{w}\|$
  $\boldsymbol{s} \leftarrow \begin{bmatrix} 0 & \beta_j & \alpha_j & \beta_{j+1} \end{bmatrix}^T$
  Olkoon $\boldsymbol{\mathcal{S}} : \mathbb{C} \to \mathbb{C}^4$ siten, että $\boldsymbol{\mathcal{S}}(z) = \kappa \boldsymbol{e}_3 z + \boldsymbol{s}\overline{z}$
  Jos $j > 2$, $\boldsymbol{\mathcal{S}}_{1:2} \leftarrow \boldsymbol{\mathcal{U}}_{j-2} \circ \boldsymbol{\mathcal{S}}_{1:2}$
  Jos $j > 1$, $\boldsymbol{\mathcal{S}}_{2:3} \leftarrow \boldsymbol{\mathcal{U}}_{j-1} \circ \boldsymbol{\mathcal{S}}_{2:3}$
  Valitaan unitaari $\boldsymbol{\mathcal{U}}_j : \mathbb{C}^2 \to \mathbb{C}^2$ siten, että $\boldsymbol{\mathcal{U}}_j \circ \boldsymbol{\mathcal{S}}_{3:4} = \begin{bmatrix} \# & 0 \end{bmatrix}^T$
  $\boldsymbol{\mathcal{S}}_{3:4} \leftarrow \boldsymbol{\mathcal{U}}_j \circ \boldsymbol{\mathcal{S}}_{3:4}$
  $\boldsymbol{\pi}_j \leftarrow (\boldsymbol{q}_j - \boldsymbol{\pi}_{j-1}\boldsymbol{\mathcal{S}}_2 - \boldsymbol{\pi}_{j-2}\boldsymbol{\mathcal{S}}_1)\boldsymbol{\mathcal{S}}_3^{-1}$
  $\boldsymbol{v}_j \leftarrow \begin{bmatrix} \boldsymbol{v}_{j-1} \\ 0 \end{bmatrix} \in \mathbb{C}^{j+1}$, $(\boldsymbol{v}_j)_{j:(j+1)} \leftarrow \boldsymbol{\mathcal{U}}_j\big((\boldsymbol{v}_j)_{j:(j+1)}\big)$
  $\boldsymbol{z} \leftarrow \boldsymbol{z} + \beta \boldsymbol{\pi}_j\big((\boldsymbol{v}_j)_j\big)$
  if $\beta_{j+1} = 0$ **then**
    lopeta algoritmi
  end if
  $\boldsymbol{q}_{j+1} \leftarrow \boldsymbol{w}/\beta_{j+1}$
end for
jäännöksen normi on $\beta|(\boldsymbol{v}_j)_{j+1}|$

Erityisesti kertomalla yhtälö $\boldsymbol{\mathcal{M}}_\kappa(\boldsymbol{z}) = \boldsymbol{b}$ vasemmalta operaattorilla $(\overline{\kappa}\boldsymbol{I} - \boldsymbol{\mathcal{M}}_0)$, saadaan $(\overline{\kappa}\boldsymbol{I} - \boldsymbol{\mathcal{M}}_0)\boldsymbol{\mathcal{M}}_\kappa = (\overline{\kappa}\boldsymbol{I} - \boldsymbol{\mathcal{M}}_0)(\kappa\boldsymbol{I} + \boldsymbol{\mathcal{M}}_0) = |\kappa|^2\boldsymbol{I} - \boldsymbol{\mathcal{M}}_0^2$, joten ratkaistavaksi yhtälöksi tulee

$$(|\kappa|^2\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}})\boldsymbol{z} = \widehat{\boldsymbol{b}}, \qquad \widehat{\boldsymbol{b}} = \overline{\kappa}\boldsymbol{b} - \boldsymbol{M}_{\#}\overline{\boldsymbol{b}}. \tag{3.54}$$

Koska $(\overline{\kappa}\boldsymbol{I} - \boldsymbol{\mathcal{M}}_0)$ on kääntyvä jos ja vain jos $(\kappa\boldsymbol{I} + \boldsymbol{\mathcal{M}}_0)$ on kääntyvä, niin (3.54) on yhtäpitävä alkuperäisen yhtälön $\boldsymbol{\mathcal{M}}_\kappa(\boldsymbol{z}) = \boldsymbol{b}$ kanssa, jos $\boldsymbol{\mathcal{M}}_\kappa$ on kääntyvä. Samoin (3.53), jos $p(\boldsymbol{\mathcal{M}}_0)$ on kääntyvä.

Matriisien GMRESia voidaan nyt käyttää näiden ratkaisuun. Suppenemisnopeus riippuu matriisista $\boldsymbol{M}_{\#}$ ja voi olla nopeampi tai hitaampi kuin edellä kuvattu reaalilineaarinen GMRES (ks. kohta 5.1). Tässä täytyy myös laskea k kertaa matriisivektoritulo $\boldsymbol{M}_{\#}$:llä jokaista uutta Krylovin aliavaruuden kantavektoria kohden.

On mielenkiintoista vertailla matriisien iteratiivisia menetelmiä tässä luvussa esiteltyihin reaalilineaarisiin menetelmiin. Kun ratkaistavana on lineaarinen yhtälöryhmä $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$, käytetään matriisien iteratiivisia menetelmiä usein seuraavasti. Hermiittiselle ja positiividefiniitille $\boldsymbol{A}$ voidaan käyttää liittogradienttimenetelmää CG (myös MINRES tulisi kyseeseen). Hermiittiselle ja indefiniitille $\boldsymbol{A}$ käytetään MINRESiä (CG:n toimivuus ei tällöin ole taattu tarkassakaan aritmetiikassa). Kun $\boldsymbol{A}$ ei ole hermiittinen, on vaihtoehtojen useita, mutta tässä luvussa on esitelty vain GMRES.

Olkoon nyt $\kappa = 1$. Vinosymmetrisellä $\boldsymbol{M}_{\#}$ on $(\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}) = (\boldsymbol{I} + \boldsymbol{M}_{\#}\boldsymbol{M}_{\#}^*)$, joka on hermiittinen ja positiividefiniitti. Siten CG soveltuu tähän. Aiemmin jo todet-

tiin kohdan 3.7.2 vinosymmetrisen $M_\#$ reaalilineaarisen menetelmän muistuttavan matriisien CG:tä. Seuraavassa on tälle menetelmälle lausetta 3.2.2 vastaava optimaalisuustulos. Tätä tapausta varten olisi myös mahdollista tehdä kohtaa 3.8.1 vastaava reaalilineaarinen MINRES vinosymmetrisille $M_\#$.

Lause 3.9.1. *Olkoon $M_\#$ vinosymmetrinen ja ratkaistavana on yhtälö $z+M_\#\overline{z} = b$. Tällöin kohdan 3.7.2 menetelmän likimääräisratkaisulle $z_k \in z_0 + \mathcal{K}_k(\mathcal{M}_1, r_0)$ pätee*

$$\|z_* - z_k\|_A \leq \|z_* - y\|_A \qquad \textit{kaikilla } y \in z_0 + \mathcal{K}_k(\mathcal{M}_1, r_0),$$

missä $A = I - M_\#\overline{M_\#}$ on hermiittinen ja positiividefiniitti, z_* on tarkka ratkaisu ja $\|z\|_A = \langle Az, z\rangle^{1/2}$.

Todistus. Kohdassa 3.7.2 nähtiin, että likimääräisratkaisu $z_k = z_0 + Q_k u_k$ saadaan ratkaisemalla yhtälö

$$u_k + T_k\overline{u}_k = \|r_0\|e_1, \tag{3.55}$$

jolla on yksikäsitteinen ratkaisu u_k. Tässä $\|r_0\|e_1 = Q_k^* r_0$ ja $T_k = Q_k^* M_\# \overline{Q_k}$. Tällöin operaattori $u \mapsto u - T_k\overline{u}$ on kääntyvä, joten (3.55) on yhtäpitävä yhtälön

$$(I - T_k\overline{T}_k)u_k = Q_k^* r_0 - T_k\overline{Q_k^* r_0}$$

kanssa. Sijoittamalla $T_k = Q_k^* M_\# \overline{Q_k}$ tästä saadaan

$$Q_k^*(I - M_\#\overline{M_\#})Q_k u_k = Q_k^*(r_0 - M_\#\overline{r_0}).$$

Väite seuraa nyt lauseesta 3.2.2. □

Symmetrisellä $M_\#$ on $(I - M_\#\overline{M_\#}) = (I - M_\# M_\#^*)$ ja tämä on mahdollisesti indefiniitti. Täten tulee kyseeseen matriisien MINRES eikä CG välttämättä toimi. Kohdan 3.7.1 symmetrisen $M_\#$ menetelmä muistuttaa kohdan 3.7.2 menetelmää ja siten matriisien CG:tä. Takeita toimivuudesta ei kuitenkaan ole mahdollisesti puuttuvan LU-hajotelman vuoksi. Sen sijaan voidaan käyttää kohdan 3.8.1 reaalilineaarista MINRESiä.

Yleisen $M_\#$ tapauksessa matriisille $(I - M_\#\overline{M_\#})$ voidaan käyttää GMRESiä ja operaattorille $z \mapsto z + M_\#\overline{z}$ kohdan 3.8 reaalilineaarista GMRESiä.

3.10 Yleisen $\mathbb{R}$-lineaarisen operaattorin menetelmät

Yleisen operaattorin $\mathcal{M}(z) = Mz + M_\#\overline{z}$ tapauksessa palataan Petrov-Galerkinin ehdosta saatuun yhtälöön (3.17). Kun z_0 on alkuarvaus ja $r_0 = b - \mathcal{M}(z_0)$ sen jäännös, voidaan ryhtyä laskemaan Arnoldin menetelmällä ortonormaaleja vektoreita kaavoilla (3.23)-(3.25) kuten algoritmissa 3.10.1. Olkoon Q_k:n sarakkeet lasketut vektorit. Etsitään ratkaisua Q_k:n sarakkeiden virittämästä aliavaruudesta ja vaaditaan jäännös kohtisuoraksi samaa aliavaruutta vastaan, jolloin saadaan yhtälö

$$Q_k^* M Q_k u_k + Q_k^* M_\# \overline{Q}_k \overline{u}_k = Q_k^* r_0\,, \tag{3.56}$$

mistä ratkaisemalla u_k saadaan likimääräisratkaisuksi $z_k = z_0 + Q_k u_k$.

Edellä kuvattu vastaa kohdan 3.7 mukaista operaattorin $\mathcal{M}_\kappa$ täyden ortogonalisaation menetelmää, mutta matriisit $Q_k^* M Q_k$ ja $Q_k^* M_\# \overline{Q}_k$ eivät nyt yksinkertaistu Hessenbergin muotoon. Kaavan (3.28) sijaan saadaan $Q_k^* M Q_k + Q_k^* M_\# \overline{Q}_k = H_k$, mitä ei voida hyödyntää. Lisäksi kantavektorit Q_k eivät välttämättä viritä Krylovin aliavaruutta $\mathcal{K}_k(\mathcal{M}, r_0)$. Tästä huolimatta tätä kantaa käyttäen yhtälö (3.56) voidaan ratkaista luvun 2 menettelyin. Näin saadaan viitteessä [1] esitetty Galerkinin approksimaatiomenetelmä `rl_Gal` yleisille $\mathbb{R}$-lineaarisille operaattoreille.

Algoritmi 3.10.1 Yksinkertainen menetelmä ortonormaalin kannan muodostamiseksi $\mathbb{R}$-operaattorille $\mathcal{M} : \mathbb{C}^n \to \mathbb{C}^n$. Laskee matriisin $Q \in \mathbb{C}^{n \times k}$, jonka sarakkeet ovat kantavektorit.

1: $q_1 \leftarrow r_0 / \|r_0\|, \quad Q \leftarrow [q_1]$
2: **for** $j = 2$ to k **do**
3: $\quad v \leftarrow \mathcal{M}(q_{j-1})$
4: $\quad w \leftarrow v - \sum_{i=1}^{j-1} \langle v, q_i \rangle \, q_i$
5: $\quad q_j \leftarrow w / \|w\|$
6: $\quad Q \leftarrow [Q \quad q_j]$
7: **end for**

Kompleksisen sisätulon sijaan voidaan myös käyttää reaalista sisätuloa, jolloin saadaan algoritmi 3.10.2. Tällöin saadut vektorit sisältyvät (reaalikertoimisiin) Krylovin aliavaruuksiin, mutta nyt ortogonalisointi ei generoi ortonormaalia kantaa kompleksisen sisätulon mielessä. Reaalisen sisätulon mielessä kohtisuorat vektorit saattavat myös olla kompleksisesti lineaarisesti riippuvat, kuten vektorit $\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$ ja $\begin{bmatrix} i & i & \cdots & i \end{bmatrix}^T$ osoittavat. Petrov-Galerkinin ehdosta (3.16) saatua yhtälöä (3.56) voidaan kuitenkin ryhtyä ratkaisemaan tälläkin Q_k vaikka menetelmän rikkoutuminen onkin mahdollista.

On huomattava, että algoritmi 3.10.2 muodostaa saman kannan kuin algoritmi 3.3.1, kunhan A on operaattoria $\mathcal{M}$ vastaava $2n \times 2n$-matriisi muodostettuna esimerkiksi käyttäen tapaa (1.3) tai (1.4). Tämän voi nähdä seuraavasti. Olkoot z, $w \in \mathbb{C}^n$, $z = x + iy$, $w = u + iv$. Tällöin

$$\operatorname{Re} \langle z, w \rangle = \operatorname{Re} \left((u^T - iv^T)(x + iy) \right) = u^T x + v^T y = \begin{bmatrix} u \\ v \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix},$$

joka on vastaava reaalinen sisätulo.

Algoritmi 3.10.2 $\mathbb{R}$-lineaarinen menetelmä kannan muodostamiseksi Krylovin aliavaruudelle $\mathcal{K}_k(\mathcal{M}, r_0)$. Laskee matriisin $Q \in \mathbb{C}^{n \times k}$, jonka sarakkeet ovat kantavektorit.

1: $q_1 \leftarrow r_0 / \|r_0\|, \quad Q \leftarrow [q_1]$
2: **for** $j = 2$ to k **do**
3: $\quad v \leftarrow \mathcal{M}(q_{j-1})$
4: $\quad w \leftarrow v - \sum_{i=1}^{j-1} \operatorname{Re}(\langle v, q_i \rangle) q_i$
5: $\quad q_j \leftarrow w / \|w\|$
6: $\quad Q \leftarrow [Q \quad q_j]$
7: **end for**

Sekä algoritmin 3.10.1 että 3.10.2 ortogonalisoinnissa voidaan (ja tietokonetoteutuksissa kannattaa) käyttää muunnettua Gram-Schmidt prosessia.

Mikäli algoritmin 3.10.2 tarjoamat kantavektorit ortonormeerataan lopuksi myös kompleksisen sisätulon suhteen, saadaan ortonormaali kanta Krylovin aliavaruuteen. Tämän tavoitteena on lisätä numeerista stabiiliutta. Näin saadaan algoritmi 3.10.3, joka on työn valvojan ehdottama [9].

Algoritmi 3.10.3 $\mathbb{R}$-lineaarinen menetelmä ortonormaalin kannan muodostamiseksi Krylovin aliavaruudelle $\mathcal{K}_k(\boldsymbol{\mathcal{M}}, \boldsymbol{r}_0)$. Laskee matriisin $\boldsymbol{Q} \in \mathbb{C}^{n \times k}$, jonka sarakkeet ovat kantavektorit.

1: $\boldsymbol{U} \leftarrow$ Algoritmi 3.10.2
2: Lasketaan (suppea) QR-hajotelma, $\boldsymbol{QR} = \boldsymbol{U}$. Hajotelman $\boldsymbol{Q}$ on tämän algoritmin tulos.

3.3.1 Esitetyistä kolmesta vaihtoehtoisesta algoritmista kantavektorimatriisin $\boldsymbol{Q}_k$ muodostamiseksi, saadaan kolme mahdollista menetelmää yhdistettäessä yhtälön (3.56) ratkaisemisen kanssa. Näiden kaikkien hankaluutena on, ettei ratkaistavaksi tulevaan yhtälöön muodostu Hessenbergin matriiseja. Sikäli nämä menetelmät jäävät epätyydyttäviksi verrattuna aikaisemmin tässä luvussa esitettyihin operaattorin $\boldsymbol{\mathcal{M}}_\kappa$ menetelmiin.

Luku 4

ILU-pohjustus

4.1 Johdanto

Olkoon $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, $\boldsymbol{b} \in \mathbb{C}^n$ ja tarkastellaan jälleen lineaarista yhtälöryhmää

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{4.1}$$

Edellisessä luvussa esiteltiin joitakin iteratiivisia ratkaisumenetelmiä, jotka pyrkivät tarkentamaan tämän yhtälön likimääräisratkaisua jokaisella kierroksellaan. Pohjustuksen tavoitteena on suppenemisen nopeuttaminen. Suppenemisnopeus riippuu matriisin $\boldsymbol{A}$ ominaisuuksista, kuten esimerkiksi GMRESille saatiin ominaisarvoista riippuva seuraus 3.4.2. Pohjustuksessa yhtälö muunnetaan sopivasti yhtäpitävään muotoon siten, että kerroinmatriisin ominaisuudet soveltuvat käytettävälle iteratiiviselle menetelmälle paremmin. GMRESin tapauksessa voidaan odottaa nopeutumista, mikäli pohjustetun kerroinmatriisin ominaisarvot kasautuvat selkeästi origosta irti olevan pisteen ympärille.

Vasemmanpuoleisessa pohjustuksessa korvataan yhtälö (4.1) yhtälöllä

$$\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b}.$$

Tähän matriisi $\boldsymbol{M}^{-1}$ saadaan varsinaisesta pohjustusmenetelmästä, joita on kehitetty lukuisia vaihdellen yleispätevyyteen pyrkivistä menetelmistä tarkoin tiettyä tehtävää varten säädettyihin menetelmiin. Tavallisesti matriisin $\boldsymbol{M}$ tulee olla jossain mielessä lähellä matriisia $\boldsymbol{A}$.

Oikeanpuoleisessa pohjustuksessa toisaalta korvataan yhtälö (4.1) yhtälöllä

$$\boldsymbol{A}\boldsymbol{M}^{-1}\boldsymbol{y} = \boldsymbol{b}.$$

Alkuperäisen yhtälön ratkaisu saadaan laskemalla lopuksi $\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{y}$. Tässäkin tulisi olla $\boldsymbol{M} \approx \boldsymbol{A}$.

Yllä mainittua matriisia $\boldsymbol{M}^{-1}$ ei yleensä lasketa eksplisiittisesti ja kerrota matriisin $\boldsymbol{A}$ kanssa, vaan näihin suhtaudutaan operaatioina $\boldsymbol{v} \mapsto \boldsymbol{M}^{-1}\boldsymbol{v}$ ja $\boldsymbol{v} \mapsto \boldsymbol{A}\boldsymbol{v}$. Iteratiivisissa menetelmissä tyypillisesti tarvitseekin vain laskea jokaisella kierroksella yhdistetyn operaattorin $\boldsymbol{v} \mapsto \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{v}$ tulos jollakin edellisestä iteraatiokierroksesta riippuvalla vektorilla $\boldsymbol{v}$. Mahdollisesti monimutkaista käänteismatriisin laskentaa ja

matriisin $\boldsymbol{A}$ harvuusrakenteen sotkevaa kertolaskua ei tarvitse tehdä. Lisäksi vaatimuksena on, että operaatio $\boldsymbol{v} \mapsto \boldsymbol{M}^{-1}\boldsymbol{v}$ on nopea suorittaa. Kokonaislaskenta-aikaa kasvattava pohjustusmenetelmä ei palvele mitään tarkoitusta, vaikka se iteraatioiden lukumäärää vähentäisikin.

Tässä luvussa käsitellään ILU-pohjustusta, joka pyrkii olemaan yleispätevä pohjustin. Edellisten lukujen tapaan esitellään aluksi matriisitapaus ja tämän jälkeen yleistetään reaalilineaarisille operaattoreille.

4.2 Matriisiyhtälöiden ILU-pohjustus

ILU-pohjustuksessa ryhdytään muodostamaan ala- ja yläkolmiomatriiseja $\boldsymbol{L}$ ja $\boldsymbol{U}$ tavallisen LU-hajotelman tapaan. Kuitenkin laskennan edetessä näistä pudotetaan pois joitakin alkiota asettamalla ne nolliksi. Tästä tuleekin menetelmän nimitys epätäydellinen LU-hajotelma (Incomplete LU). Lopputulokselle pätee vain suunnilleen $\boldsymbol{A} \approx \boldsymbol{LU}$.

Edellä olevassa johdannossa oleva matriisi $\boldsymbol{M}$ on nyt $\boldsymbol{M} = \boldsymbol{LU}$, jota ei kuitenkaan tarvitse muodostaa eksplisiittisesti tietokoneen muistiin. Käänteisoperaatio $\boldsymbol{v} \mapsto \boldsymbol{M}^{-1}\boldsymbol{v}$ voidaan laskea yhtälöistä

$$\begin{aligned} \boldsymbol{Ly} &= \boldsymbol{v} \\ \boldsymbol{Ux} &= \boldsymbol{y} \end{aligned}$$

eteen- ja taaksepäin sijoituksin. Pohjustusoperaation tulos on $\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{v}$. Kun matriiseja $\boldsymbol{L}$ ja $\boldsymbol{U}$ muodostettaessa alkioita pudotetaan pois riittävästi, tulee operaatiosta $\boldsymbol{v} \mapsto \boldsymbol{U}^{-1}\boldsymbol{L}^{-1}\boldsymbol{v}$ nopea. Toisaalta, jos alkioita pudotetaan pois liikaa, tulee approksimaatiosta $\boldsymbol{A} \approx \boldsymbol{LU}$ huono eikä menetelmä nopeuta iteraation suppenemista.

ILU-hajotelman poispudotettavat alkiot voidaan valita eri strategioin ja näistä saadaan kokonainen perhe pohjustusmenetelmiä. Seuraavassa esitellään kaksi mahdollisuutta. Huomattakoon, ettei kummassakaan käytetä tuentaa luvun 2 tapaan. Näin hajotelman laskeminen nopeutuu, mutta ei ole numeerisesti stabiili. Lisäksi tukialkio saattaa olla nolla. Jälkimmäisestä ongelmasta pääsisi eroon korvaamalla tukialkion jollakin nollasta poikkeavalla luvulla, mutta tämä huonontaa $\boldsymbol{LU} \approx \boldsymbol{A}$ tarkkuutta. Tuenta on mahdollista myös ILU-hajotelmille, mutta sen käsittely on tämän työn ulkopuolella.

4.2.1 ILU-hajotelma ennaltavalitulla nollakuviolla

Olkoon $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ ja $Z \subset \left\{(i,j) \in \mathbb{Z}\times\mathbb{Z} \,\middle|\, 1 \le i,j \le n\,,\, i \ne j\right\}$. Tulkitaan seuraavassa, että joukko Z on ILU-hajotelman ns. nollakuvio. Jos indeksipari (i,j) kuuluu joukkoon Z, niin tätä indeksiparia vastaava alkio pudotetaan pois. Mikäli (i,j) on aidossa alakolmiossa $(i > j)$, niin asetetaan matriisin $\boldsymbol{L}$ alkio $l_{ij} = 0$. Jos taas (i,j) on aidossa yläkolmiossa $(i < j)$, niin asetetaan $u_{ij} = 0$ matriisissa $\boldsymbol{U}$. Matriisien lävistäjäalkioita ei voida asettaa nolliksi, koska $\boldsymbol{L}$:n ja $\boldsymbol{U}$:n täytyy olla kääntyviä.

Aluksi asetetaan kaikki $\boldsymbol{A}$:n alkiot $a_{ij} = 0$, jos (i,j) kuuluu nollakuvioon. Sitten

ryhdytään laskemaan tämän $\boldsymbol{A}$:n LU-hajotelmaa Gaussin eliminoinnein. Tämä suoritetaan muuten tuttuun tapaan, mutta rivien yhteenlaskussa nollakuvioon kuuluvia (aluksi nolliksi asetettuja) alkioita ei ylikirjoiteta. Näin saadaan algoritmi 4.2.1. Tässä matriisit $\boldsymbol{L}$ ja $\boldsymbol{U}$ on pidetty erillään. Yleensä LU-toteutuksissa hajotelma muodostetaan suoraan matriisin $\boldsymbol{A}$ päälle, jolloin sen yläkolmioon muodostuu $\boldsymbol{U}$:n alkiot ja aitoon alakolmioon $\boldsymbol{L}$:n aito alakolmio eikä $\boldsymbol{L}$:n yksiköksi tiedettyä lävistäjää tarvitse tallentaa.

Algoritmi 4.2.1 Laskee ILU-hajotelman matriisit $\boldsymbol{L} = (l_{ij}), \boldsymbol{U} = (u_{ij})$ matriisille $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ nollakuviolla Z.

```
1: L ← I
2: U ← A
3: u_ij ← 0 kaikille (i, j) ∈ Z.
4: for i = 1 to n do
5:    for k = i + 1 to n do
6:       if (k, i) ∉ Z then
7:          l_ki ← u_ki / u_ii
8:          u_kr ← u_kr − l_ki u_ir kaikilla r > i s.e. (k, r) ∉ Z
9:       end if
10:      u_ki ← 0
11:   end for
12: end for
```

Algoritmi 4.2.2 laskee saman ILU-hajotelman, mutta kahden uloimman silmukan järjestystä on vaihdettu. Vastaava `MATLAB`-funktio `ilu_v2` löytyy liitteestä B. Silmukoiden järjestyksen vaihto vastaa Gaussin eliminoinnissa sitä, että se suoritetaan valmiiksi järjestyksessä rivi kerrallaan ensimmäisestä aloittaen. Järjestyksen vaihdolla saadaan paremmin harvoille matriiseille soveltuva algoritmi, koska ainoastaan työn alla olevaa riviä muokataan kerrallaan. Rivin laskennan valmistuttua se voidaan tallentaa tietokoneen muistiin harvan matriisin tietorakenteen muotoon ja siirtyä käsittelemään seuraavaa riviä. Ennaltavalitulla nollakuviolla ei tällä ole laskemisen kannalta suurta merkitystä. Seuraavassa kohdassa esitettävälle ILUT-hajotelmalle silmukoiden vaihto on kuitenkin tärkeä muunnelma ja sitä käytetään tästä lähtien yksinomaan (ILUT:lle laskennan lopputulos myös riippuu silmukoiden järjestyksestä).

Tähän mennessä nollakuvion valinnasta ei olla vielä sanottu mitään. Seuraava määritelmä tekee yksinkertaisen valinnan.

Määritelmä 4.2.1. *Matriisin $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ ILU(0)-hajotelmaksi kutsutaan ILU-hajotelmaa, missä nollakuvioksi on valittu $\boldsymbol{A}$:n nollakuvio. Tarkemmin $(i, j) \in Z \Leftrightarrow a_{ij} = 0$.*

Liitteessä olevien `MATLAB`-koodien avulla ILU(0)-hajotelman voi laskea suorittamalla `ilu_v2(A, A)`.

4.2.2 ILU-hajotelma mukautetulla nollakuviolla

Tarkastellaan edellisen kohdan algoritmia 4.2.2. Siinä alkioiden pudotuksia tapahtuu kahdessa paikassa; sekä `for k`-silmukassa että `for j`-silmukassa. Alkioiden pu-

Algoritmi 4.2.2 Laskee ILU-hajotelman matriisit $\boldsymbol{L} = (l_{ij}), \boldsymbol{U} = (u_{ij})$ matriisille $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ nollakuviolla Z.

```
1: L ← I
2: U ← A
3: for i = 1 to n do
4:   for k = 1 to i − 1 do
5:     if (i, k) ∈ Z then
6:       l_ik ← 0
7:     else
8:       l_ik ← u_ik/u_kk
9:       u_i* ← u_i* − l_ik u_k*
10:    end if
11:    u_ik ← 0
12:  end for
13:  for j = i + 1 to n do
14:    if (i, j) ∈ Z then
15:      u_ij ← 0
16:    end if
17:  end for
18: end for
```

dotus perustuu ennaltavalittuun nollakuvioon Z. Mukautetulla nollakuviolla tarkoitetaan tässä sellaista alkioiden pudotusmenettelyä, joka ottaa huomioon niiden (itseis)arvojen suuruuden. Hajotelmassa säilytetään itseiarvoltaan suuret alkiot pois pudotettavien ollessa pieniä, jolloin approksimaatiosta $\boldsymbol{A} \approx \boldsymbol{LU}$ saadaan parempi kuin suuruudet huomiotta jättävissä ennaltavalitun nollakuvion menetelmissä.

Matriisien $\boldsymbol{L}$ ja $\boldsymbol{U}$ alkioista pois pudotetaan sellaiset, joiden suhteellinen koko (verrattuna $\boldsymbol{A}$:n vastaavan koko rivin normiin) on riittävän pieni. Olkoon $\epsilon > 0$ ja a_{i*} matriisin $\boldsymbol{A}$ rivi i. Tällöin hajotelman laskettu alkio l_{ij} pudotetaan pois mikäli $|l_{ij}| < \epsilon \|a_{i*}\|$. Samoin laskettu u_{ij} pudotetaan pois, jos $|u_{ij}| < \epsilon \|a_{i*}\|$. Tässä normi on tavallinen 2-normi, mutta myös muiden normien käyttö on mahdollista.

Lisäksi valitaan argumentti p, joka määrää sekä $\boldsymbol{L}$:n aidossa alakolmiossa että $\boldsymbol{U}$:n aidossa yläkolmiossa kullakin rivillä olevien nollasta poikkeavien alkioiden suurimman määrän. Alkioista säilytetään p itseisarvoltaan suurinta ja loput pudotetaan pois. Tällä saadaan lisää hallittavuutta $\boldsymbol{L}$:n ja $\boldsymbol{U}$:n harvuusrakenteeseen.

Tälläisella menettelyllä muodostettua hajotelmaa kutsutaan ILUT:ksi ja se on esitetty algoritmina 4.2.3. Vastaava `MATLAB`-funktio `ilut` löytyy liitteestä B.

4.3 $\mathbb{R}$-lineaaristen yhtälöiden ILU-pohjustus

Olkoon $\mathcal{M} : \mathbb{C}^n \to \mathbb{C}^n$ reaalilineaarinen operaattori ja $\boldsymbol{b} \in \mathbb{C}^n$. Yhtälölle

$$\mathcal{M}(\boldsymbol{z}) = \boldsymbol{b} \tag{4.2}$$

on edellisessä luvussa esitetty iteratiivisia menetelmiä, joita matriisiyhtälöiden tapaan lähdetään nyt pohjustamaan. Tavoitteena on iteraatiokierrosten ja kokonaislaskenta-ajan pienentäminen käyttäen nopeasti laskettavaa pohjustinta.

Algoritmi 4.2.3 ILUT(ϵ, p). Laskee ILU-hajotelman matriisit $\boldsymbol{L} = (l_{ij}), \boldsymbol{U} = (u_{ij})$ matriisille $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ käyttäen mukautettua nollakuviota. Argumentti ϵ määrää alkioiden suuruustoleranssin suhteessa rivikokoon. Tuloksena olevissa $\boldsymbol{L}$:ssä ja $\boldsymbol{U}$:ssa on lävistäjäalkioiden lisäksi korkeintaan p nollasta poikkeavaa alkiota kullakin rivillä.

1: $\boldsymbol{L} \leftarrow \boldsymbol{I}$
2: $\boldsymbol{U} \leftarrow \boldsymbol{A}$
3: **for** $i = 1$ to n **do**
4: $\quad m \leftarrow \|u_{i*}\|$
5: $\quad$ **for** $k = 1$ to $i - 1$ **do**
6: $\quad\quad c \leftarrow u_{ik}/u_{kk}$
7: $\quad\quad$ **if** $|c| < \epsilon m$ **then**
8: $\quad\quad\quad l_{ik} \leftarrow 0$
9: $\quad\quad$ **else**
10: $\quad\quad\quad l_{ik} \leftarrow c$
11: $\quad\quad\quad u_{i*} \leftarrow u_{i*} - cu_{k*}$
12: $\quad\quad$ **end if**
13: $\quad\quad u_{ik} \leftarrow 0$
14: $\quad$ **end for**
15: $\quad$ **for** $j = i + 1$ to n **do**
16: $\quad\quad$ **if** $|u_{ij}| < \epsilon m$ **then**
17: $\quad\quad\quad u_{ij} \leftarrow 0$
18: $\quad\quad$ **end if**
19: $\quad$ **end for**
20: $\quad$ **if** $i > p + 1$ **then**
21: $\quad\quad$ Säilytetään alkioista $l_{i,1}, \ldots, l_{i,i-1}$ p itseisarvoltaan suurinta; muut asetetaan nolliksi.
22: $\quad$ **end if**
23: $\quad$ **if** $i < n - p$ **then**
24: $\quad\quad$ Säilytetään alkioista $u_{i,i+1}, \ldots, u_{i,n}$ p itseisarvoltaan suurinta; muut asetetaan nolliksi.
25: $\quad$ **end if**
26: **end for**

Olkoon $\mathcal{P} : \mathbb{C}^n \to \mathbb{C}^n$ jonkin pohjustusmenetelmän antama operaattori siten, että $\mathcal{P} \approx \mathcal{M}$. Vasemmanpuoleisessa pohjustuksessa ratkotaan yhtälön (4.2) sijaan yhtälöä

$$\mathcal{P}^{-1}\mathcal{M}(z) = \mathcal{P}^{-1}(b).$$

Matriisitapausta vastaten, oikeanpuoleinen pohjustus on

$$\mathcal{M}\mathcal{P}^{-1}(w) = b,$$

mistä alkuperäisen yhtälön (4.2) ratkaisu saadaan laskemalla $z = \mathcal{P}^{-1}(w)$.

On huomattava, ettei siirretty antilineaarinen operaattori $\mathcal{M}_\kappa(z) = \kappa z + M_\# \overline{z}$ edellä kuvatulla pohjustuksella säilytä muotoaan. Olkoon $\mathcal{P}^{-1}(z) = Sz + S_\# \overline{z}$, jolloin laskemalla saadaan

$$\mathcal{P}^{-1}\mathcal{M}_\kappa(z) = (\kappa S + S_\# \overline{M_\#})z + (SM_\# + \overline{\kappa} S_\#)\overline{z},$$

mistä nähdään ettei lineaarinen osa yleensä ole muotoa $\widehat{\kappa} I$. Edellisessä luvussa tälle tärkeälle erikoistapaukselle esiteltiin reaalilineaarinen GMRES, mutta sen pohjustaminen on tällä hetkellä avoin ongelma. Pohjustus koskeekin tässä työssä ainoastaan yleiselle operaattorille kohdassa 3.10 esiteltyjä menetelmiä. Näidenkin menetelmien hankaluutena on, että lineaarisella ja antilineaarisella osalla täytyy kyetä operoimaan erikseen. Kun operaattorin $\mathcal{M}$ matriisit M ja $M_\#$ ovat tunnettuja, täytyisi tämän jälkeen pohjustetusta operaattorista $\mathcal{P}^{-1}\mathcal{M}$ vielä erotella vastaavat matriisit, mikä ei yleensä ole mielekästä. Tämän vuoksi joudutaan käyttämään seuraavia kaavoja, jotka pätevät mille tahansa operaattorille $\mathcal{N}(z) = Nz + N_\# \overline{z}$,

$$\begin{aligned} Nz &= \tfrac{1}{2}\big(\mathcal{N}(z) - i\mathcal{N}(iz)\big), \\ N_\# \overline{z} &= \tfrac{1}{2}\big(\mathcal{N}(z) + i\mathcal{N}(iz)\big). \end{aligned}$$

Valitsemalla $\mathcal{N} = \mathcal{P}^{-1}\mathcal{M}$ nähdään, että pohjustetun operaattorin osilla voidaan iteratiivista menetelmää varten aina laskea, mutta tähän tarvitaan kaksi täyttä operaattori-vektori tuloa. Algoritmia 3.10.3 vastaavan menetelmän pohjustaminen on erityisen työlästä, koska pohjustusoperaatiot täytyy laskea yllä kuvatulla tavalla ensin algoritmia 3.10.2 ajettaessa ja sen jälkeen uudestaan QR-hajotelmasta saatujen vektoreiden kanssa.

$\mathbb{R}$-lineaarisessa ILU-pohjustuksessa operaattoriksi $\mathcal{P}$ valitaan $\mathcal{P} = \mathcal{L}\mathcal{U}$, missä operaattorit $\mathcal{L} : \mathbb{C}^n \to \mathbb{C}^n$ ja $\mathcal{U} : \mathbb{C}^n \to \mathbb{C}^n$ muodostavat $\mathbb{R}$-lineaarisen epätäydellisen LU-hajotelman. Perustana on luvussa 2 esitetty reaalilineaarinen LU-hajotelma.

4.3.1 $\mathbb{R}$-lineaarinen ILU-hajotelma ennaltavalitulla nollakuviolla

Olkoon $\mathcal{M} : \mathbb{C}^n \to \mathbb{C}^n$, $\mathcal{M}(z) = Mz + M_\# \overline{z}$, reaalilineaarinen operaattori ja

$$Z \subset \big\{(i,j) \in \mathbb{Z} \times \mathbb{Z} \,\big|\, 1 \le i,j \le n\,,\; i \ne j\big\}$$

nollakuvio. Ryhdytään muodostamaan $\mathcal{M}$:n LU-hajotelmaa kohdan 2.1 tavalla, mutta asettamalla nollaksi nollakuviossa olevat alkiot. Menetelmä on täysin analoginen kohdan 4.2.1 matriisitapauksen kanssa. Matriisien tilalla voidaan ajatella manipuloitavan muodossa (1.1) esitettyjä reaalilineaarisia operaattoreita.

Aluksi asetetaan nolliksi ne matriisien $\boldsymbol{M}$ ja $\boldsymbol{M}_{\#}$ alkiot, joiden indeksipari (i,j) kuuluu nollakuvioon. Sitten suoritetaan Gaussin eliminointeja luvun 2 tapaan, mutta rivejä yhteenlaskiessa nollakuvioon kuuluvaa indeksiparia (i,j) vastaavaa (alussa nollaksi asetettua) skalaarioperaattoria ei koskaan ylikirjoiteta. Algoritmi 4.3.1 laskee ILU-hajotelman vastaten matriisitapauksen algoritmia 4.2.2, missä eliminointi suoritetaan valmiiksi rivi kerrallaan. Vastaava `MATLAB`-funktio `rl_ilu` löytyy liitteestä B.

Algoritmi 4.3.1 Laskee $\mathbb{R}$-ILU-hajotelman operaattorit $\boldsymbol{\mathcal{L}}(\boldsymbol{z}) = \boldsymbol{L}\boldsymbol{z} + \boldsymbol{L}_{\#}\overline{\boldsymbol{z}}$, $\boldsymbol{\mathcal{U}}(\boldsymbol{z}) = \boldsymbol{U}\boldsymbol{z} + \boldsymbol{U}_{\#}\overline{\boldsymbol{z}}$ operaattorille $\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}}$ nollakuviolla Z.

$\boldsymbol{L} \leftarrow \boldsymbol{I}$, $\boldsymbol{L}_{\#} \leftarrow 0$
$\boldsymbol{U} \leftarrow \boldsymbol{M}$, $\boldsymbol{U}_{\#} \leftarrow \boldsymbol{M}_{\#}$
for $i = 1$ to n **do**
  for $k = 1$ to $i-1$ **do**
    if $(i,k) \in Z$ **then**
      $l_{ik} \leftarrow 0$, $l_{\#,ik} \leftarrow 0$
    else
      $$\begin{bmatrix} l_{ik} & l_{\#,ik} \end{bmatrix} \leftarrow \begin{bmatrix} u_{ik} & u_{\#,ik} \end{bmatrix} \begin{bmatrix} u_{kk} & u_{\#,kk} \\ \overline{u_{\#}}_{kk} & \overline{u}_{kk} \end{bmatrix}^{-1}$$
      $$u_{i*} \leftarrow u_{i*} - \begin{bmatrix} l_{ik} & l_{\#,ik} \end{bmatrix} \begin{bmatrix} u_{k*} \\ \overline{u_{\#}}_{k*} \end{bmatrix}$$
      $$u_{\#,i*} \leftarrow u_{\#,i*} - \begin{bmatrix} l_{ik} & l_{\#,ik} \end{bmatrix} \begin{bmatrix} u_{\#,k*} \\ \overline{u}_{k*} \end{bmatrix}$$
    end if
    $u_{ik} \leftarrow 0$, $u_{\#,ik} \leftarrow 0$
  end for
  for $j = i+1$ to n **do**
    if $(i,j) \in Z$ **then**
      $u_{ij} \leftarrow 0$, $u_{\#,ij} \leftarrow 0$
    end if
  end for
end for

Yksinkertainen nollakuvio saadaan seuraavasta määritelmästä.

Määritelmä 4.3.1. *Operaattorin $\boldsymbol{\mathcal{M}}$ $\mathbb{R}$-ILU(0):ksi kutsutaan hajotelmaa, jossa nollakuvioksi on valittu matriisien $\boldsymbol{M}$ ja $\boldsymbol{M}_{\#}$ yhteinen nollakuvio. Tarkemmin $(i,j) \in Z \Leftrightarrow \big((\boldsymbol{M})_{ij} = 0$ ja $(\boldsymbol{M}_{\#})_{ij} = 0\big)$.*

Kun `M` ja `Ma` ovat `MATLAB`in harvoja matriiseja, funktion `rl_ilu` avulla $\mathbb{R}$-ILU(0):n voi laskea suorittamalla `rl_ilu(M, Ma, max(spones(M), spones(Ma)))`.

4.3.2 $\mathbb{R}$-lineaarinen ILU-hajotelma mukautetulla nollakuviolla

Reaalilineaarinen ILUT, jota tämän jälkeen kutsutaan lyhyesti $\mathbb{R}$-ILUT:ksi, saadaan edellä olevan kohdan koodista vastaavin muutoksin kuin matriisien tapauksessa kohdassa 4.2.2.

Olkoon $\epsilon > 0$ ja m_{i*}, $m_{\#,i*}$ matriisien $\boldsymbol{M}$ ja $\boldsymbol{M}_{\#}$ rivit i. Merkitään reaalilineaarisen LU-hajotelman operaattoreita $\boldsymbol{\mathcal{L}}(\boldsymbol{z}) = \boldsymbol{L}\boldsymbol{z} + \boldsymbol{L}_{\#}\overline{\boldsymbol{z}}$ ja $\boldsymbol{\mathcal{U}}(\boldsymbol{z}) = \boldsymbol{U}\boldsymbol{z} + \boldsymbol{U}_{\#}\overline{\boldsymbol{z}}$. Alkiot

l_{ij} ja $l_{\#,ij}$ pudotetaan pois, jos $|l_{ij}| + |l_{\#,ij}| < \epsilon(\|l_{i*}\|_1 + \|l_{\#,i*}\|_1)$. Samoin u_{ij} ja $u_{\#,ij}$ pudotetaan pois, jos $|u_{ij}| + |u_{\#,ij}| < \epsilon(\|u_{i*}\|_1 + \|u_{\#,i*}\|_1)$. Näissä on käytetty 1-normia, mutta toisenlainenkin normin valinta on mahdollista.

Jälleen, argumentti p määrä määrää sekä $\boldsymbol{\mathcal{L}}$:n aidossa alakolmiossa että $\boldsymbol{\mathcal{U}}$:n aidossa yläkolmiossa kullakin rivillä olevien nollasta poikkeavien skalaarioperaattoreiden suurimman määrän. Skalaareista säilytetään p normiltaan suurinta ja loput pudotetaan pois.

Tuloksena on algoritmi 4.3.2, joka vastaa matriisien algoritmia 4.2.3. Liitteessä B on vastaava `MATLAB`-funktio nimellä `rl_ilut`.

Yleisestä reaalimatriisiesityksestä (1.4) nähdään, että reaalilineaariset ILU-hajotelmat voidaan nähdä matriisien lohko-ILU-hajotelmina, missä lohkot ovat 2×2-kokoisia. Täten näiden tehokkuuden voidaan olettaa olevan samaa luokkaa tunnettujen matriisien lohkohajotelmiin perustuvien pohjustusmenetelmien kanssa. Tehtyjä numeerisia kokeita esitetyille ILU-hajotelmille käydään läpi seuraavassa luvussa.

Algoritmi 4.3.2 $\mathbb{R}$-ILUT(ϵ, p). Laskee $\mathbb{R}$-ILUT-hajotelman operaattorit $\boldsymbol{\mathcal{L}}(\boldsymbol{z}) = \boldsymbol{L}\boldsymbol{z} + \boldsymbol{L}_{\#}\overline{\boldsymbol{z}}$, $\boldsymbol{\mathcal{U}}(\boldsymbol{z}) = \boldsymbol{U}\boldsymbol{z} + \boldsymbol{U}_{\#}\overline{\boldsymbol{z}}$ operaattorille $\boldsymbol{\mathcal{M}}(\boldsymbol{z}) = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}}$ käyttäen mukautettua nollakuviota. Argumentti ϵ määrää alkioiden suuruustoleranssin suhteessa rivikokoon. Tuloksena olevissa $\boldsymbol{L}, \boldsymbol{L}_{\#}$:ssä ja $\boldsymbol{U}, \boldsymbol{U}_{\#}$:ssa on lävistäjäalkioiden lisäksi korkeintaan p nollasta poikkeavaa alkiota kullakin rivillä.

$\boldsymbol{L} \leftarrow \boldsymbol{I}$, $\boldsymbol{L}_{\#} \leftarrow 0$
$\boldsymbol{U} \leftarrow \boldsymbol{M}$, $\boldsymbol{U}_{\#} \leftarrow \boldsymbol{M}_{\#}$
for $i = 1$ to n **do**
  $m \leftarrow \|u_{i*}\|_1 + \|u_{\#,i*}\|_1$
  for $k = 1$ to $i - 1$ **do**
    $\begin{bmatrix} c & c_{\#} \end{bmatrix} \leftarrow \begin{bmatrix} u_{ik} & u_{\#,ik} \end{bmatrix} \begin{bmatrix} u_{kk} & u_{\#,kk} \\ \overline{u_{\#}}_{kk} & \overline{u}_{kk} \end{bmatrix}^{-1}$
    if $|c| + |c_{\#}| < \epsilon m$ **then**
      $l_{ik} \leftarrow 0$, $l_{\#,ik} \leftarrow 0$
    else
      $l_{ik} \leftarrow c$, $l_{\#,ik} \leftarrow c_{\#}$
      $u_{i*} \leftarrow u_{i*} - \begin{bmatrix} c & c_{\#} \end{bmatrix} \begin{bmatrix} u_{k*} \\ \overline{u_{\#}}_{k*} \end{bmatrix}$
      $u_{\#,i*} \leftarrow u_{\#,i*} - \begin{bmatrix} c & c_{\#} \end{bmatrix} \begin{bmatrix} u_{\#,k*} \\ \overline{u}_{k*} \end{bmatrix}$
    end if
    $u_{ik} \leftarrow 0$, $u_{\#,ik} \leftarrow 0$
  end for
  for $j = i + 1$ to n **do**
    if $|u_{ij}| + |u_{\#,ij}| < \epsilon m$ **then**
      $u_{ij} \leftarrow 0$, $u_{\#,ij} \leftarrow 0$
    end if
  end for
  if $i > p + 1$ **then**
    Säilytetään alkioista $l_{i,1}, l_{\#,i,1}, \ldots, l_{i,i-1}, l_{\#,i,i-1}$ ne, joiden indeksi ovat samat kuin p:n suurimman listassa $|l_{i,1}| + |l_{\#,i,1}|, \ldots, |l_{i,i-1}| + |l_{\#,i,i-1}|$; muut asetetaan nolliksi.
  end if
  if $i < n - p$ **then**
    Säilytetään alkioista $u_{i,i+1}, u_{\#,i,i+1}, \ldots, l_{i,n}, l_{\#,i,n}$ ne, joiden indeksi ovat samat kuin p:n suurimman listassa $|u_{i,i+1}| + |u_{\#,i,i+1}|, \ldots, |l_{i,n}| + |l_{\#,i,n}|$; muut asetetaan nolliksi.
  end if
end for

Luku 5

Numeerisia kokeita

Tässä luvussa esitetään suoritettujen numeeristen kokeiden koejärjestelyt ja tulokset. Kokeet jakaantuvat kahteen osaan. Aluksi tarkastellaan ja vertaillaan luvun 3 GMRES-menetelmiä, jonka jälkeen siirrytään luvun 4 pohjustusmenetelmien kokeisiin. Kaikki numeerinen laskenta tehtiin `MATLAB`in versiolla 7.6.0.324 (R2008a).

5.1 GMRES kokeita

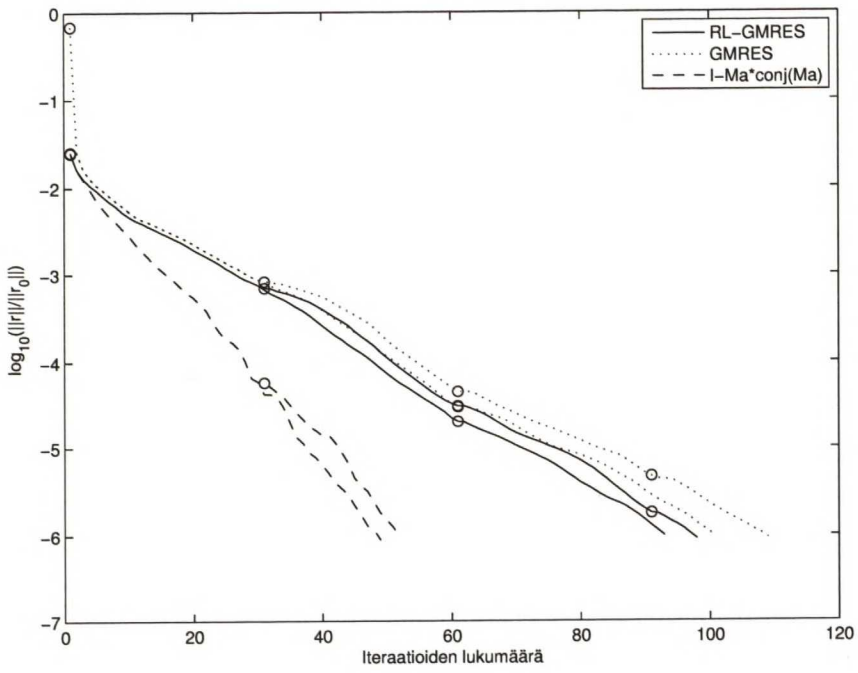
Kokeissa ratkaistiin yhtälöryhmä $\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, missä vektori $\boldsymbol{b}$ koostuu luvuista 1. Menetelminä olivat reaalilineaarinen GMRES (RL-GMRES, kohta 3.8) ja matriisien GMRES (kohta 3.4) sekä kaksinkertaiseksi kirjoitetulle reaaliselle yhtälöryhmälle (kohdan 3.8 loppu) että yhtälölle (3.54).

Kokeet suoritettiin `MATLAB`illa ja ohjelmakoodit perustuivat viitteen [1] `rl_GMRES`-funktioon. Matriisien GMRES toteutuksena käytettiin `rl_GMRES`ista muokattua versiota, joten molempien toteutustapa oli erittäin samankaltainen. Kokeissa matriiseja $\boldsymbol{M}_{\#}$ oli kolmea tyyppiä. Merkitään $\boldsymbol{R}_n^{(j)} \in \mathbb{R}^{n\times n}$ matriisia, jonka alkiot ovat valittu tasaisesti satunnaisesti väliltä $(0,1)$. Nämä matriisit muodostettiin `MATLAB`illa käyttäen $\boldsymbol{R}_n^{(j)} = \text{rand(n)}$ ja ne vaihtuivat satunnaisesti jokaisessa testiajossa.

Ensimmäisessä kokeessa oli $\boldsymbol{M}_{\#} = (\boldsymbol{I}_{500} + \boldsymbol{R}_{500}^{(1)} + i\boldsymbol{R}_{500}^{(2)})/10 \in \mathbb{C}^{500\times 500}$. Tyypillisen ajon jäännösvektoreiden normit $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_0\|$ on piirretty kuvaan 5.1 $\log_{10}$-skaalassa. Toisessa kokeessa $\boldsymbol{M}_{\#} = (5\boldsymbol{I}_{100} + \boldsymbol{R}_{100}^{(1)} + i\boldsymbol{R}_{100}^{(2)}) \in \mathbb{C}^{100\times 100}$. Jäännösvektoreiden normit on piirretty kuvaan 5.2. Viimeisessä kokeessa ei käytetty satunnaismatriiseja vaan valittiin $\boldsymbol{M}_{\#} = \text{diag}(2,3,\ldots,101) \in \mathbb{C}^{100\times 100}$, mistä saatiin kuva 5.3. Alkuarvauksena oli $\boldsymbol{z}_0 = 0$ kaikissa kokeissa.

Tasapuolisen vertailun vuoksi yhtälön (3.54) jäännösvektoreiden normit laskettiin kaavalla $\|\boldsymbol{r}_k\| = \|\boldsymbol{b} - \boldsymbol{\mathcal{M}}_1(\boldsymbol{z}_k)\|$ jokaisella iteraatiokierroksella. Tulokset eivät kuitenkaan juurikaan poikenneet käytännöllisen iteroinnin kaavasta $\|\widehat{\boldsymbol{r}}_k\| = \|(\boldsymbol{I} - \boldsymbol{\mathcal{M}}_0)(\boldsymbol{b} - \boldsymbol{\mathcal{M}}_1(\boldsymbol{z}_k))\|$.

Kuvista nähdään, että RL-GMRES suppenee kaksinkertaista reaalista systeemia hieman nopeammin lukuunottamatta viimeistä koetta, jossa suppeneminen on yhtä nopeaa. Yhtälö (3.54) vaatii vain suunnilleen puolet edellisten iteraatioista lukuunotta-



Kuva 5.1: Iteraatioiden jäännöksen suhteellinen virhe, kun $\kappa = 1$ ja $\boldsymbol{M}_{\#} =$ (eye(500) + rand(500) + i*rand(500))/10 ja vektori $\boldsymbol{b}$ koostuu luvuista 1. Kuvaan on piirretty sekä 30 että 60 iteraation välein uudelleenkäynnistetyt GMRESit. Käynnistyskohdat on merkitty symbolilla ○.

matta viimeistä koetta, jossa se ei suppene. Toisaalta tämän yhtälön ratkaisu vaatii kahden matriisi-vektoritulon laskemista $\boldsymbol{M}_{\#}$:lla.

5.2 GMRES impedanssitomografiassa

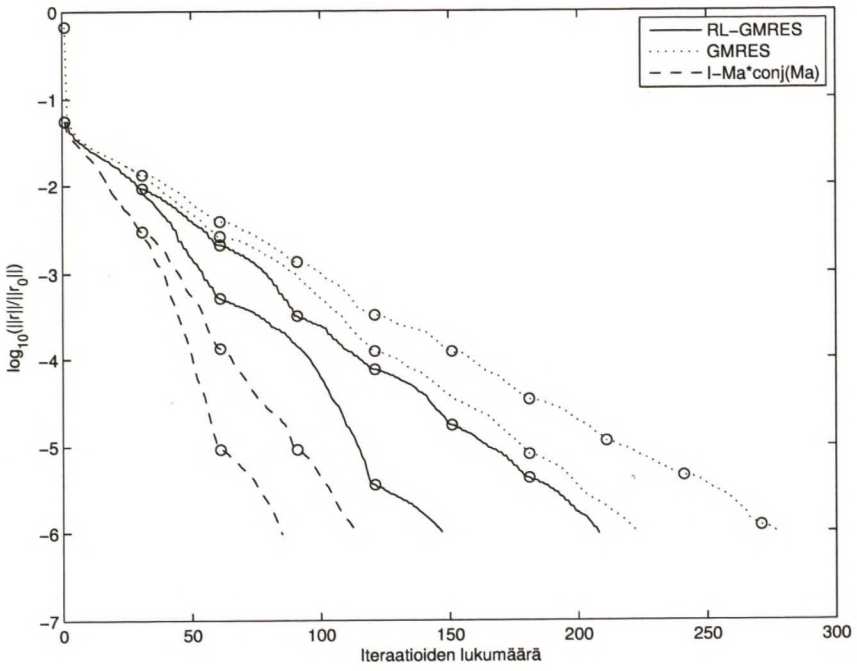
Edellä olevissa GMRES-kokeissa nähtiin, että suppenemisnopeus riippuu hyvin paljon teennäisesti valitusta matriisista $\boldsymbol{M}_{\#}$. Käytännön suorituskyvyn selvittämiseksi kokeita suoritettiin myös impedanssitomografiassa esiintyvillä yhtälöillä ja simuloidulla datalla.

Olkoon Ω tason $\mathbb{R}^2$ avoin yksikkökiekko ja tarkastellaan siellä johtavuusyhtälön reuna-arvotehtävää

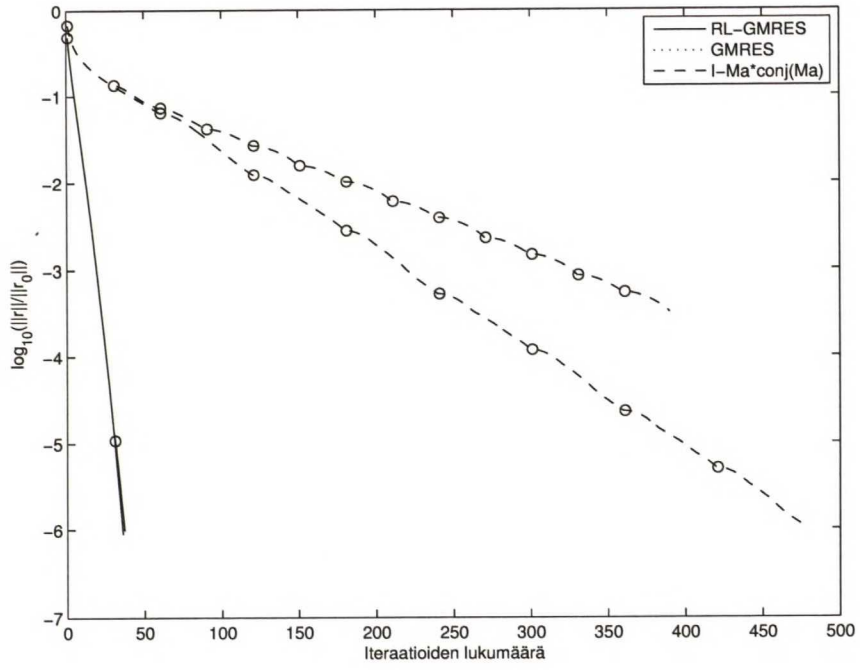
$$\begin{cases} \nabla \cdot \gamma(\boldsymbol{x}) \nabla u(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in \partial\Omega, \end{cases}$$

missä sähkönjohtavuus γ on vakioiden rajoittama $0 < c \leq \gamma \leq C < \infty$. Kun (sähkö)potentiaali f kiekon reunalla on annettu ja johtavuus γ tunnetaan, ratkaisu u on potentiaali koko kiekossa. Tällöin voidaan laskea kiekon reunalla virrantiheys $\gamma \frac{\partial u}{\partial \nu}|_{\partial\Omega}$ funktio, missä ν on yksikkönormaalivektori ulospäin. Tunnetun johtavuuden γ perusteella voidaan täten määritellä kuvaus Λ_γ reunalla annetuista potentiaaleista f reunan virrantiheyksiin, $\Lambda_\gamma(f) = \gamma \frac{\partial u}{\partial \nu}|_{\partial\Omega}$.

Sähköisessä impedanssitomografiassa ratkaistavana on inversio-ongelma, missä ku-



Kuva 5.2: Iteraatioiden jäännöksen suhteellinen virhe, kun $\kappa = 1$ ja $\boldsymbol{M}_{\#} =$ 5*eye(100) + rand(100) + i*rand(100) ja vektori $\boldsymbol{b}$ koostuu luvuista 1. Kuvaan on piirretty sekä 30 että 60 iteraation välein uudelleenkäynnistetyt GMRESit. Käynnistyskohdat on merkitty symbolilla ○.



Kuva 5.3: Iteraatioiden jäännöksen suhteellinen virhe, kun $\kappa = 1$ ja $\boldsymbol{M}_{\#} =$ diag$(2, 3, \ldots, 101)$. Kuvaan on piirretty sekä 30 että 60 iteraation välein uudelleenkäynnistetyt GMRESit. Käynnistyskohdat on merkitty symbolilla ○. RL-GMRES ja GMRES viivat ovat tässä päällekkäin.

vaus Λ_γ on annettu ja tätä vastaava johtavuus γ on ratkaistava. Varsin yleisin oletuksin voidaan osoittaa, että γ on yksikäsitteinen [7]. Tätä ennen vaativammin oletuksin yksikäsitteisyyden on todistanut mm. Brown ja Uhlmann (mm. γ:lla täytyy olla ensimmäiset osittaisderivaatat). Kuten [5], tässä esityksessä käytetään Brown-Uhlmannin todistuksessa käytettyä menetelmää γ:n rekonstruoimiseen. Johtavuuden γ uudelleenrakentaminen jakautuu kahteen vaiheeseen. Ensimmäisessä vaiheessa kuvauksen Λ_γ perusteella lasketaan nk. sirontamuunnos ja toisessa vaiheessa sirontamuunnoksesta rakennetaan johtavuus γ. Jälkimmäinen vaihe tapahtuu ratkaisemalla $\overline{\partial}$-yhtälöitä (5.1). Tässä tarkoituksena on kokeilla luvun 3 menetelmiä, joten kokonaisen inversiotehtävän ratkaisemisen sijaan keskitytään tähän jälkimmäiseen vaiheeseen. Tällöin on sallittua hieman oikaista ja laskea sirontamuunnos suoraan annetusta johtavuudesta γ. Brown-Uhlmannin menetelmässä tämäkin voidaan tehdä ratkaisemalla $\overline{\partial}$-yhtälöitä.

5.2.1 $\overline{\partial}$-yhtälön ratkaiseminen

Tarkastellaan seuraavaa nk. $\overline{\partial}$-yhtälöä

$$\overline{\partial}_z w(z) = -T(z)\overline{w(z)}, \qquad \lim_{|z|\to\infty} w(z) = 1 \tag{5.1}$$

missä kerroinfunktiolla $T : \mathbb{R}^2 \to \mathbb{C}$ on kompakti kantaja. Tästä on ratkaistava $w : \mathbb{R}^2 \to \mathbb{C}$ asymptoottisella ehdolla $\lim_{|z|\to\infty} w(z) = 1$. Kompleksiluku $z = z_1 + iz_2$ samaistetaan tason $\mathbb{R}^2$ pisteen (z_1, z_2) kanssa ja käytetään derivaatta-operaattoreita

$$\partial_z = \frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial z_1} - i\frac{\partial}{\partial z_2}\right), \qquad \overline{\partial}_z = \frac{\partial}{\partial \bar{z}} = \frac{1}{2}\left(\frac{\partial}{\partial z_1} + i\frac{\partial}{\partial z_2}\right),$$

missä $z = (z_1, z_2)$. Yhtälö (5.1) ja w:n asymptoottinen ehto ovat yhtäpitäviä integraaliyhtälön

$$w(z) = 1 - \frac{1}{\pi}\int_{\mathbb{R}^2} \frac{T(z')}{z - z'}\overline{w(z')}\, dz_1' dz_2' \tag{5.2}$$

kanssa, missä $g(z) = 1/(\pi z)$ on operaattorin $\overline{\partial}_z$ Greenin funktio. Voidaan kirjoittaa myös lyhyemmin

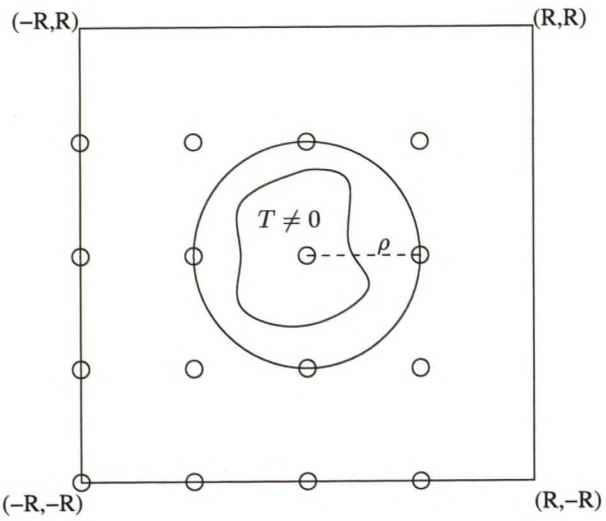
$$w + g * (T\overline{w}) = 1, \tag{5.3}$$

missä $*$ tarkoittaa funktioiden konvoluutiota.

Numeerista laskentaa varten diskretoidaan integraaliyhtälö (5.2) viitteen [5] tavalla. Olkoon $\overline{B}(0, \rho)$ origokeskinen ρ-säteinen suljettu kiekko siten, että T:n kantaja sisältyy siihen. Olkoon $R \geq 2\rho$ ja asetetaan $S_R = [-R, R) \times [-R, R)$ (ks. kuva 5.4). Yhtälön (5.2) ratkaisu w kiekossa $\overline{B}(0, \rho)$ riippuu Greenin funktion g arvoista vain kiekossa $\overline{B}(0, 2\rho)$, joten g voidaan haluttaessa määritellä uudelleen tämän kiekon ulkopuolella. Olkoon

$$K(z) = \begin{cases} \dfrac{1}{\pi z}, & \text{kun } |z| \leq R, \\ 0, & \text{kun } |z| > R. \end{cases} \tag{5.4}$$

Katkaisu voitaisiin tehdä myös sileästi valitsemalla $R = 2\rho + \epsilon$ jollakin pienellä $\epsilon > 0$ ja asettamalla $K(z) = g(z)\chi(z)$, missä χ on sileä, $0 \leq \chi \leq 1$ ja saa arvon 1 kiekossa $\overline{B}(0, 2\rho)$ ja 0 kiekon $\overline{B}(0, R)$ ulkopuolella. Tässä esityksessä käytetään vain katkaisua (5.4).



Kuva 5.4: Keskellä avoin kiekko $B(0,\rho)$ ja $R = 2\rho$. Pienet ympyrät esittävät 4×4 hilaa ($m = 2$).

Tarkastellaan periodista integraaliyhtälöä

$$w(z) = 1 - \int_{S_R} K(z - z')T(z')\overline{w(z')}\, dz_1' dz_2' \qquad (z \in S_R), \tag{5.5}$$

missä K ja T on ensin rajoitettu neliöön S_R ja sitten jatkettu $2R$-periodisiksi funktioiksi koko tasoon. Tämän ratkaisu w on niin ikään $2R$-periodinen funktio.

Periodisen yhtälön (5.5) ratkaisulle $\widetilde{w}$ ja alkuperäisen yhtälön (5.2) ratkaisulle w pätee $\widetilde{w}(z) = w(z)$ kaikilla $z \in B(0,\rho)$. Kun periodisesta yhtälöstä (5.5) on saatu ratkaistua $\widetilde{w}$, voitaisiin se laajentaa kiekosta $B(0,\rho)$ yhtälön (5.2) ratkaisuksi koko $\mathbb{R}^2$:een laskemalla

$$w(z) = 1 - \frac{1}{\pi} \int_{B(0,\rho)} \frac{T(z')}{z - z'} \overline{\widetilde{w}(z')}\, dz_1' dz_2' \qquad (z \in \mathbb{R}^2).$$

Tässä työssä ratkaisun kiinnostava osuus kuitenkin sijaitsee kiekossa $B(0,\rho)$ eikä tätä laajennusta tarvitse laskea.

Ryhdytään sitten ratkaisemaan yhtälöä (5.5). Olkoon m positiivinen kokonaisluku ja asetetaan $N = 2^m$, $h = 2R/N$. Muodostetaan hila pisteistä jh, missä $j \in \mathbb{Z}_N^2$

$$\mathbb{Z}_N^2 = \left\{ (j_1, j_2) \in \mathbb{Z}^2 \,\middle|\, -\frac{N}{2} \le j_1, j_2 < \frac{N}{2} \right\}. \tag{5.6}$$

Tämä on $N \times N$-hila, jonka tapaus $m = 2$ on piirretty kuvaan 5.4. Tällainen N:n valinta sopii erityisen hyvin jäljempänä esiteltävän nopean Fourier-muunnoksen käyttöön.

Kun $j \in \mathbb{Z}^2$, merkitään

$$T_j = T(jh), \tag{5.7}$$

$$w_j = w(jh), \tag{5.8}$$

$$K_j = \begin{cases} 0, & \text{kun } j_1 \equiv j_2 \equiv 0 \pmod N, \\ K(jh), & \text{muutoin.} \end{cases} \tag{5.9}$$

Funktion K diskretoinnissa asetettiin nolla singulariteettipisteissä. Singulariteettipisteet ovat integroituvia, joten tästä aiheutuva diskretointivirhe pienenee hilaa tihentäessä. Koska funktiot T, w ja K ovat $2R$-periodisia, ovat diskretoidut kaksiindeksiset jonot T_j, w_j ja K_j N-periodisia molempien indeksien suhteen.

Diskretoidaan seuraavaksi yhtälössä (5.5) esiintyvä integraali muotoon

$$h^2 \sum_{k \in \mathbb{Z}_N^2} K_{j-k} T_k \overline{w_k}. \tag{5.10}$$

Tässä $K_{j-k} = K((j-k)h) = K(jh - kh)$, kun $j \neq k$. Summa (5.10) on periodisten jonojen konvoluutio, joka voidaan merkitä lyhyemmin $h^2(K_j) * (T_j \overline{w_j})$. Yhtälö (5.5) on diskretoitu nyt muotoon

$$w_j + h^2(K_j) * (T_j \overline{w_j}) = 1 \qquad (j \in \mathbb{Z}_N^2).$$

Konvoluutiota ei kannata laskea suoraan kaavasta (5.10) vaan käyttäen nopeaa Fourier-muunnosta (FFT:stä voi lukea esim. kirjasta [10]). Merkitään $\mathcal{F}$:llä kaksiulotteista diskreettiä Fourier-muunnosta, jolloin edellä olevasta yhtälöstä tulee

$$w_j + h^2 \mathcal{F}^{-1}\big(\mathcal{F}(K_j) \cdot \mathcal{F}(T_j \overline{w_j})\big) = 1, \qquad (j \in \mathbb{Z}_N^2), \tag{5.11}$$

missä $\cdot$ tarkoittaa Fourier-kertoimien kertomista keskenään alkioittain.

Matriisiyhtälöksi tämä voidaan kirjoittaa pakkaamalla ratkaistavat w_j vektoriksi $\boldsymbol{u} \in \mathbb{C}^{N^2}$ asettamalla $u_{(j_1+N/2)+N(j_2+N/2)+1} = w_{j_1,j_2}$, missä $j = (j_1, j_2) \in \mathbb{Z}_N^2$. Tämän pakkauksen myötä summa (5.10) voidaan esittää matriisien avulla. Yhtälöksi saadaan $\boldsymbol{u} + \boldsymbol{M}_{\#}\overline{\boldsymbol{u}} = \boldsymbol{1}$, missä $\boldsymbol{1} \in \mathbb{C}^{N^2}$ koostuu luvuista 1 ja $\boldsymbol{M}_{\#} = \boldsymbol{C}\boldsymbol{D}$, missä $\boldsymbol{C}, \boldsymbol{D} \in \mathbb{C}^{N^2 \times N^2}$. Matriisi $\boldsymbol{D}$ on diagonaalimatriisi koostuen alkioista T_j ja alkioista K_j koostuvalla matriisilla $\boldsymbol{C}$ kertominen suorittaa konvoluution. Tämä yhtälö ratkeaa operaattorille $\boldsymbol{\mathcal{M}}_\kappa$ luvussa 3 esitellyin menettelyin.

Trigonometrinen kollokaatio

Kerrotaan yhtälö (5.5) puolittain $\overline{T(z)}$:lla, jolloin

$$\overline{T(z)}w(z) = \overline{T(z)} - \overline{T(z)} \int_{S_R} K(z - z')T(z')\overline{w(z')}\, dz_1' dz_2' \qquad (z \in S_R).$$

Merkitsemällä $v(z) = \overline{T(z)}w(z)$ saadaan periodinen integraaliyhtälö

$$v(z) = \overline{T(z)} - \overline{T(z)} \int_{S_R} K(z - z')\overline{v(z')}\, dz_1' dz_2'. \tag{5.12}$$

Kun tämä ratkaistaan ja lasketaan funktio w kaavasta

$$w(z) = 1 - \int_{S_R} K(z - z')\overline{v(z')}, dz_1' dz_2',$$

niin w on yhtälön (5.5) ratkaisu. Diskretoidaan (5.12) käyttäen trigonometristä kollokaatiota viitteen [6] tapaan.

Määritellään ortonormaali kanta avaruuteen $L^2(S_R)$ funktioilla

$$\varphi_j(z) = \frac{1}{2R} \exp\left(i\pi j \cdot \frac{z}{R}\right), \qquad j \in \mathbb{Z}^2, \tag{5.13}$$

ja koostukoon avaruus $\mathcal{T}_N$ trigonometrisistä polynomeista $v_N = \sum_{j\in\mathbb{Z}_N^2} c_j\varphi_j$, missä $c_j \in \mathbb{C}$. Kun v on $2R$-periodinen funktio, määritellään sen trigonometrinen interpolaatio $Q_N v$ vaatimalla

$$Q_N v \in \mathcal{T}_N, \quad (Q_N v)(jh) = v(jh), \quad j \in \mathbb{Z}_N^2.$$

Yhtälö (5.12) diskretoidaan nyt muotoon

$$v_N = Q_N(\overline{T}) - Q_N(\overline{T}K\overline{v_N}), \tag{5.14}$$

missä $(Kv)(z) = \int_{S_R} K(z-z')v(z')\,dz_1'dz_2'$ ja on ratkaistava $v_N \in \mathcal{T}_N$. Nähdään, että

$$K\varphi_j = \widehat{K}(j)\varphi_j \qquad (j \in \mathbb{Z}^2),$$

missä $\widehat{K}(j) = \int_{S_R} K(z)\varphi_{-j}(z)\,dz_1dz_2$ ovat funktion K Fourier-kertoimet.

Trigonometrisellä polynomilla $v_N \in \mathcal{T}_N$ on seuraavat kaksi mahdollista esitystapaa

$$v_N(z) = \sum_{k\in\mathbb{Z}_N^2} \widehat{v}_N(k)\varphi_k(z),$$
$$v_N(z) = \sum_{j\in\mathbb{Z}_N^2} v_N(jh)\varphi_{N,j}(z),$$

missä $\varphi_{N,j} \in \mathcal{T}_N$ toteuttaen $\varphi_{N,j}(kh) = \delta_{j,k}$ (Kroneckerin δ) ja $j,k \in \mathbb{Z}_N^2$. Ensimmäinen näistä on Fourier-kertoimien avulla esitetty ja jälkimmäinen hilapisteissä olevien arvojen avulla. Funktiot $\varphi_{N,j}$ saadaan kaavasta

$$\varphi_{N,j}(z) = \frac{1}{N^2} \sum_{k\in\mathbb{Z}_N^2} \exp\left(i\pi k \cdot \left(\frac{x}{R} - \frac{2j}{N}\right)\right).$$

Esityksestä toiseen voidaan siirtyä diskreetin Fourier-muunnoksen avulla. Laskemalla saadaan

$$\widehat{v}_N(k) = \int_{S_R} v_N(z)\varphi_{-j}(z)\,dz_1dz_2 = \frac{2R}{N^2} \sum_{j\in\mathbb{Z}_N^2} v_N(jh) \exp\left(-i\pi k \cdot \frac{2j}{N}\right). \tag{5.15}$$

Arvot hilapisteissä lasketaan diskreetillä käänteismuunnoksella

$$v_N(jh) = \frac{1}{2R} \sum_{k\in\mathbb{Z}_N^2} \widehat{v}_N(k) \exp\left(i\pi k \cdot \frac{2j}{N}\right). \tag{5.16}$$

Muodostetaan sitten yhtälön (5.14) matriisimuoto. Kun ratkaisun $v_N \in \mathcal{T}_N$ arvot $v_N(jh)$, $j \in \mathbb{Z}_N^2$, tunnetaan, niin $v_N = \sum_{k\in\mathbb{Z}_N^2} \widehat{v}_N(k)\varphi_k$, missä kertoimet $\widehat{v}_N(k)$ saadaan kaavasta (5.15). Joukon (5.6) lisäksi määritellään

$$\widetilde{\mathbb{Z}}_N^2 = \left\{(j_1,j_2) \in \mathbb{Z}^2 \,\middle|\, -\frac{N}{2} < j_1, j_2 \le \frac{N}{2}\right\}.$$

Sijoitetaan ja lasketaan

$$\begin{aligned}\overline{T}K\overline{v_N} &= \overline{T}\sum_{k\in\mathbb{Z}_N^2}\overline{\widehat{v}_N(k)}\widehat{K}(-k)\varphi_{-k} = \overline{T}\sum_{k\in\mathbb{Z}_N^2}\widehat{v}_N(-k)\widehat{K}(-k)\varphi_{-k}\\ &= \overline{T}\sum_{k\in\widetilde{\mathbb{Z}}_N^2}\widehat{K}(k)\widehat{v}_N(k)\varphi_k,\end{aligned}$$

joten matriisimuodoksi saadaan

$$v_j + \overline{T_j}\mathcal{F}_h^{-1}\big(\widehat{K}(j)\cdot\mathcal{F}_h(\overline{v_j})\big) = \overline{T_j}, \qquad (j\in\mathbb{Z}_N^2), \tag{5.17}$$

missä $v_j = v_N(jh)$ ja $\mathcal{F}_h, \mathcal{F}_h^{-1}$ ovat diskreetit Fourier-muunnokset (5.15) ja (5.16). Kun v_j pakataan vektoriksi $\boldsymbol{y}$ edellisen kohdan lopussa olevaa tapaa käyttäen, niin tämä yhtälö on matriisimuotoa

$$\boldsymbol{y} + \overline{\boldsymbol{D}}\boldsymbol{C}\overline{\boldsymbol{y}} = \overline{\boldsymbol{D}}\boldsymbol{1}. \tag{5.18}$$

Kun tästä ratkaistaan $\boldsymbol{y}$ ja lasketaan $\boldsymbol{u} = \boldsymbol{1} - \boldsymbol{C}\overline{\boldsymbol{y}}$, niin nähdään, että $\boldsymbol{u}$ toteuttaa yhtälön

$$\boldsymbol{u} + \boldsymbol{C}\boldsymbol{D}\overline{\boldsymbol{u}} = \boldsymbol{1}. \tag{5.19}$$

Toisaalta kertomalla yhtälö (5.19) puolittain vasemmalta $\overline{\boldsymbol{D}}$:llä ja ottamalla uudeksi muuttujaksi $\boldsymbol{y} = \overline{\boldsymbol{D}}\boldsymbol{u}$ päädytään yhtälöön (5.18). Nämä ovat siten yhtäpitävät. Yhtälöön (5.19) päädyttiin myös edellisessä kohdassa. Erona on, että yhtälön (5.11) Fourier-kertoimet $\mathcal{F}(K_j)$ on laskettu diskreetillä Fourier-muunnoksella diskretoinnista (5.9) ja tämän paikalla on nyt tarkasti laskettavissa olevat kertoimet $\widehat{K}(j)$. Lasketaan nämä seuraavaksi. Huomautettakoon, ettei yhtälössä (5.17) ole tekijää h^2 yhtälön (5.11) tapaan. Tämä tekijä sisältyy nyt kertoimiin $\widehat{K}(j)$.

Katkaistun Greenin funktion Fourier-kertoimet

Olkoon $j = (j_1, j_2) \neq (0,0)$. Derivoimalla nähdään, että $\overline{\partial}_z\varphi_j(z) = \lambda_j^{-1}\varphi_j(z)$, missä $\lambda_j = \left(\frac{\pi}{2R}(-j_2 + ij_1)\right)^{-1}$, joten

$$\begin{aligned}\widehat{K}(j) &= \int_{S_R} K(z)\varphi_{-j}(z)\,dz_1dz_2\\ &= \int_{B(0,R)}\frac{1}{\pi z}\varphi_{-j}(z)\,dz_1dz_2 = -\lambda_j\int_{B(0,R)}\frac{1}{\pi z}\overline{\partial}_z\varphi_{-j}(z)\,dz_1dz_2\\ &= -\lambda_j\lim_{\delta\to 0}\int_{B(0,R)\setminus B(0,\delta)}\frac{1}{\pi z}\overline{\partial}_z\varphi_{-j}(z)\,dz_1dz_2\end{aligned}$$

$$\begin{aligned}= -\lambda_j\lim_{\delta\to 0}\Bigg\{&\left(\int_{S(0,R)} - \int_{S(0,\delta)}\right)\left(\frac{1}{\pi z}\varphi_{-j}(z)\frac{z}{2|z|}\right)ds\\ &- \int_{B(0,R)\setminus B(0,\delta)}\left(\overline{\partial}_z\frac{1}{\pi z}\right)\varphi_{-j}(z)\,dz_1dz_2\Bigg\},\end{aligned}$$

missä on käytetty Gauss-Green lausetta. Jälkimmäisin integraali on nolla, koska $1/(\pi z)$ on operaattorin $\overline{\partial}$ Greenin funktio. Saadaan

$$\widehat{K}(j) = \lambda_j\left(\varphi_{-j}(0) - \frac{1}{2\pi R}\int_{S(0,R)}\varphi_{-j}(z)\,ds\right).$$

Lasketaan

$$\begin{aligned}\int_{S(0,R)} e^{i\pi j\cdot z/R}\,ds &= R\int_{S(0,1)} e^{i\pi j\cdot z}\,ds = R\int_{S(0,1)} e^{i\pi|j|z_1}\,ds\\ &= 2R\int_{-1}^{1} e^{i\pi|j|z_1}(1-z_1^2)^{-1/2}\,dz_1\\ &= 2R\int_{-1}^{1}\cos(\pi|j|z_1)(1-z_1^2)^{-1/2}\,dz_1 = 2\pi R J_0(\pi|j|),\end{aligned}$$

missä Besselin funktiolle J_0 käytetty kaava löytyy esim. kirjasta [16]. Täten

$$\widehat{K}(j) = \frac{\lambda_j}{2R}\big(1 - J_0(\pi|j|)\big).$$

Kun $j = (0,0)$, nähdään helposti $\widehat{K}(0) = \int_{B(0,R)} \frac{1}{\pi z}\,dz_1dz_2 = 0$. Fourier-kertoimiksi on näin saatu

$$\widehat{K}(j) = \begin{cases}\dfrac{1 - J_0(\pi|j|)}{\pi(-j_2 + ij_1)}, & \text{kun } j \neq (0,0),\\ 0, & \text{kun } j = (0,0).\end{cases}$$

5.2.2 Brown-Uhlmann menetelmä

Seuraavassa esitetään vain ratkaistavat yhtälöt ja laskentakaavat, katso [5] ja sen viitteet tarkempaan matemaattiseen kuvaukseen. Olkoon sitten johtavuusfunktio γ annettu kiekossa Ω ja oletetaan, että $\gamma = 1$ lähellä kiekon reunaa. Lasketaan (eräänlainen potentiaali)

$$q = -\gamma^{-1/2}\partial_z\gamma^{1/2}. \tag{5.20}$$

Koska oletetaan, että $\gamma = 1$ lähellä kiekon reunaa $\partial\Omega$, voidaan q laajentaa sileästi koko tasoon asettamalla $q(z) = 0$ kaikilla $z \in \mathbb{C}\setminus\Omega$. Ratkaistaan kaikki seuraavat $\overline{\partial}$-yhtälöt parametrina $k \in \mathbb{C}$

$$\overline{\partial}_z m_\pm(z,k) = \pm q(z)e(z,-k)\overline{m_\pm(z,k)}, \qquad \lim_{|z|\to\infty} m_\pm(z,k) = 1, \tag{5.21}$$

missä $e(z,k) = \exp(i(zk+\overline{zk})) = \exp(2i\,\mathrm{Re}(zk))$. Lasketaan $m_1(z,k) = \frac{1}{2}\big(m_+(z,k)+m_-(z,k)\big)$ ja potentiaalin q sirontamuunnos kaavasta

$$S(k) = \frac{-i}{\pi}\int_\Omega e(z,k)\overline{q(z)}m_1(z,k)\,dz_1dz_2. \tag{5.22}$$

Johtavuus γ uudelleenrakennetaan sirontamuunnoksesta S ratkaisemalla $\overline{\partial}$-yhtälöt parametrina $z \in \Omega$

$$\overline{\partial}_k\widetilde{m}^+(z,k) = \overline{S(-k)}e(z,-k)\overline{\widetilde{m}^+(z,k)}, \qquad \lim_{|k|\to\infty}\widetilde{m}^+(z,k) = 1. \tag{5.23}$$

Johtavuus saadaan sitten laskemalla

$$\gamma(z) = \big(\mathrm{Re}(\widetilde{m}^+(z,0))\big)^2. \tag{5.24}$$

5.2.3 Kokeet

Kokeissa toistettiin viitteen [5, kohta 5.1] järjestely. Ensiksi valmisteltiin johtavuusfunktio γ yksikkökiekkoon. Taustajohtavuus on 1, sydämen johtavuus on 2, keuhkojen johtavuus on 0.33 ja toisessa keuhkossa olevan poikkeaman johtavuus 1.33. Kuvassa 5.5 on tämä alkuperäinen johtavuus jo uudelleenrakennettuna. Potentiaali q laskettiin käyttäen kaavaa (5.20) ja numeerista derivointia.

Yhtälöt (5.21) ratkaistiin käyttäen kohdan 5.2.1 menetelmää. Neliöön $[-2,2)\times[-2,2)$ muodostettiin 128×128-hila muuttujan z arvoille ja neliöön $[-40,40) \times [-40,40)$ niin ikään 128×128-hila muuttujan k arvoille. Sirontamuunnos laskettiin kaavasta (5.22) numeerisella integroinnilla. Koska sirontamuunnos S ei poikkea nollasta vain rajoitetussa joukossa, se täytyy katkaista seuraavaa vaihetta varten. Tästä aiheutuu systemaattinen virhe, mutta tämä jätetään huomiotta, koska S on $|k|$:n kasvaessa nopeasti vaimeneva. Edellä neliö $[-40,40) \times [-40,40)$ on jo valittu tätä katkaisua varten eikä $S(k)$ arvoja laskettu, kun $|k| > 20$, vaan asetettiin nolliksi.

Laskettu sirontamuunnos tallennettiin ja seuraavan vaiheen johtavuuden rekonstruointiin käytetyt iteraatiot ja laskenta-ajat mitattiin eri menetelmin. Jokaisessa kokeessa ratkaistiin $\overline{\partial}$-yhtälöt (5.23) kaikille z:n arvoille neliössä $[-1,1)\times[-1,1)$ 32×32-hilassa lukuunottamatta hilapisteitä, joille $|z| > 1$. Jokaisessa hilapisteessä ratkaistiin reaalilineaarinen yhtälö $\boldsymbol{u} + \boldsymbol{M}_{\#}\overline{\boldsymbol{u}} = \mathbf{1}$ iteratiivisin menetelmin ja iterointi pysäytettiin, kun jäännös $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_0\| = \|\mathbf{1} - \boldsymbol{u}_k - \boldsymbol{M}_{\#}\overline{\boldsymbol{u}}_k\|/\|\boldsymbol{r}_0\| < 10^{-12}$.

Jokaisen kokeen tuloksena oli kuva 5.5 ja mittaukset eri menetelmin on kirjattu taulukkoon 5.1. Laskenta-ajat mitattiin `MATLAB`in `TIC-TOC` komennoilla kahdessa eri ajossa. Taulukossa $\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ tarkoittaa yhtälön (3.54) ratkaisemista matriisien GMRESillä ja $\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}\boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ yhtälön (3.53) ratkaisua arvolla $k = 4$. Näille kahdelle suoritettiin kokeet myös käyttäen epäkäytännöllistä jäännöksen normia $\|\boldsymbol{r}_k\| = \|\boldsymbol{b} - \boldsymbol{\mathcal{M}}_1(\boldsymbol{u}_k)\|$. Nämä on merkitty J:llä. Tämä tehtiin, jotta iterointi tulisi pysäytettyä samassa tarkkuudessa muiden menetelmien kanssa. Kuten taulukosta nähdään, ei tällä ole iteraatioiden määrään juurikaan vaikutusta. Laskenta-aika J:llä merkityissä kohdissa johtuu kohtuuttomasta ylimääräisestä laskennasta. Kuvaan 5.6 on piirretty yhden $\overline{\partial}$-yhtälön ratkaisun suhteelliset jäännökset.

Vertailun vuoksi laskenta suoritettiin myös seuraavalla yksinkertaisella kiintopisteiteraatiolla

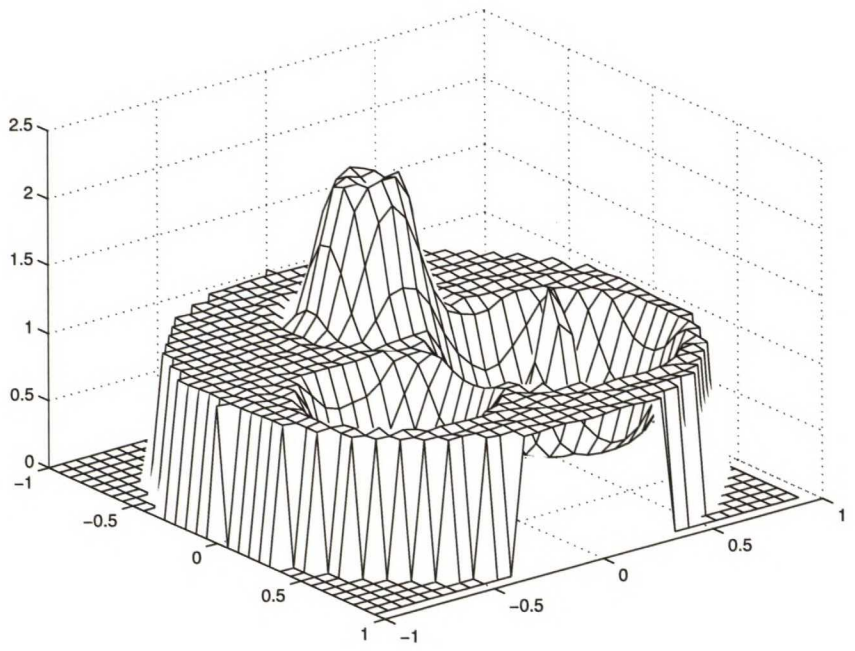
$$\boldsymbol{u}_{k+1} = -\boldsymbol{M}_{\#}\overline{\boldsymbol{u}}_k + \mathbf{1}.$$

Tämän iterointi pysäytettiin, kun $\|\mathbf{1} - \boldsymbol{u}_{k+1} - \boldsymbol{M}_{\#}\overline{\boldsymbol{u}}_k\|/\|\boldsymbol{r}_0\| < 10^{-12}$. Kuten taulukosta nähdään, iteraatioiden lukumäärä on suurempi kuin GMRES-menetelmissä, mutta laskenta-aika on tästä huolimatta hyvä.

5.2.4 $\overline{\partial}$-yhtälö satunnaisella kerroinfunktiolla

Edellisen kohdan kiintopisteiteraation ja GMRES-menetelmän menestys matriisille $\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}\boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ antaa aihetta epäillä, että tämä johtuu matriisin $\boldsymbol{M}_{\#}$ normin pienuudesta.

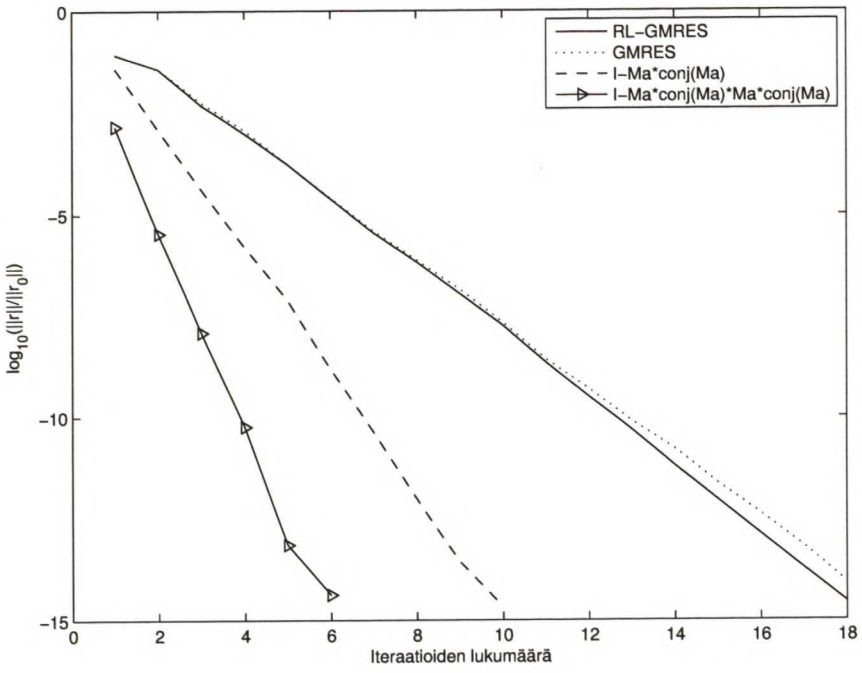
Seuraavissa kokeissa yhtälö (5.1) ratkaistiin neliössä $[-40,40) \times [-40,40)$ 256×256-hilassa ja kerroinfunktion T arvot hilapisteissä $|z| < 20$ asetettiin tasaisesti satunnai-



Kuva 5.5: Johtavuus yksikköympyrässä.

Menetelmä	Iteraatiot	Aika (s)
RL-GMRES	12192	105.1
	12192	104.9
GMRES	12500	120.9
	12500	119.0
$\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$	6742	69.0
	6742	68.3
$\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ J	6744	97.7
	6744	98.5
$\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}\boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$	3756	61.3
	3756	61.1
$\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}\boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ J	3764	76.1
	3764	75.3
Kiintopiste	15991	55.1
	15991	55.2

Taulukko 5.1: Sirontamuunnoksesta lasketun rekonstruoinnin aika.



Kuva 5.6: Iteraatioiden jäännöksen suhteellinen virhe, kun $\overline{\partial}$-yhtälö on ratkaistu hilan keskipisteessä ($z_1 = z_2 = 0$).

Menetelmä	0.01	0.1	0.5	1.0	5.0	10.0	20.0
RL-GMRES	5	10	25	39	160	313	625
GMRES	5	10	25	41	179	355	713
$\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ J	3	5	13	22	80	155	352
$\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}\boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ J	2	4	36	> 960			
Kiintopiste	7	75	∞	∞			

Taulukko 5.2: $\overline{\partial}$-yhtälö ratkaistuna T:llä, jonka arvot ovat tasaisesti satunnaisia väliltä 0 ylimmällä rivillä olevaan arvoon. Arvot ovat iteraatioiden lukumääriä ja merkinnällä ∞ menetelmä ei supennut.

sesti 0:n ja annetun ylärajan välillä `MATLAB`in `rand`-funktiolla. Muissa hilapisteissä T asetettiin nollaksi. Kaikki GMRESit uudelleenkäynnistettiin 60 iteraation välein ja pysäytettiin, kun jäännöksen normi oli $< 10^{-6}$. Tulokset ovat taulukossa 5.2, mistä nähdään kiintopisteiteraation ja matriisin $\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}\boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ GMRESin suppenemisen hidastuvan rajusti T:n kasvaessa.

On huomattava, että $\boldsymbol{I} - \boldsymbol{M}_{\#}\overline{\boldsymbol{M}_{\#}}$ säilytti nopeussuhteensa RL-GMRESiin ja kaksinkertaiseen reaaliseen GMRESiin nähden.

5.3 Aaltoyhtälö

Tarkastellaan seuraavaa kaksiulotteisen aaltoyhtälön alkuarvo- ja reuna-arvotehtävää alueessa $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$

$$\begin{cases} u_{tt}(\boldsymbol{x},t) = \Delta u(\boldsymbol{x},t)\,, & \boldsymbol{x} \in \Omega\,,\ t > 0\,, \\ u(\boldsymbol{x},t) = 0\,, & \boldsymbol{x} \in \partial\Omega\,,\ t \geq 0\,, \\ u(\boldsymbol{x},0) = f(\boldsymbol{x})\,,\ u_t(\boldsymbol{x},0) = g(\boldsymbol{x})\,, & \boldsymbol{x} \in \Omega. \end{cases}$$

Ratkaistaan tämä numeerisesti käyttämällä aluksi differenssiapproksimaatioita paikkamuuttujan suhteen. Olkoon $n_1, n_2 \in \mathbb{Z}^+$ ja asetetaan $h_1 = \frac{1}{n_1}$, $h_2 = \frac{1}{n_2}$. Merkitään $u_{(i,j)}(t) = u(ih_1, jh_2, t)$, $0 \leq i \leq n_1+1$, $0 \leq j \leq n_2+1$. Reunaehdoista johtuen $u_{(0,j)}(t) = u_{(n_1+1,j)}(t) = u_{(i,0)}(t) = u_{(i,n_2+1)}(t) = 0$. Määritellään $n \times n$-matriisi

$$\boldsymbol{\Delta}_h = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix},$$

missä $h = \frac{1}{n}$. Tämä on itse asiassa yksidimensioisen Laplacen operaattorin (toinen derivaatta) differenssiapproksimaatio, kun molemmissa päissä on Dirichlet'n reunaehto 0.

Kaksidimensioista tapausta varten järjestetään tuntemattomat $u_{(i,j)}(t)$, $1 \leq i \leq n_1$, $1 \leq j \leq n_2$, pystyvektoriksi. Merkitään $v_{i+(j-1)n_1}(t) = u_{(i,j)}(t)$. Kroneckerin tulon avulla muodostettua matriisia

$$\boldsymbol{\Delta}_{h_1,h_2} = \boldsymbol{I}_{n_2} \otimes \boldsymbol{\Delta}_{h_1} + \boldsymbol{\Delta}_{h_2} \otimes \boldsymbol{I}_{n_1}$$

käyttämällä tulee tehtäväksi ratkaista differentiaaliyhtälösysteemi

$$\boldsymbol{v}''(t) = \boldsymbol{\Delta}_{h_1,h_2}\boldsymbol{v}(t)\,, \qquad \boldsymbol{v}(0) = \boldsymbol{f}_{h_1,h_2}\,,\ \boldsymbol{v}'(0) = \boldsymbol{g}_{h_1,h_2}\,,$$

missä $(\boldsymbol{f}_{h_1,h_2})_{i+(j-1)n_1} = f(ih_1, jh_2)$ ja $(\boldsymbol{g}_{h_1,h_2})_{i+(j-1)n_1} = g(ih_1, jh_2)$. Kirjoitetaan tämä ensimmäisen kertaluvun systeemiksi. Merkitään $\boldsymbol{w}(t) = \boldsymbol{v}'(t)$, jolloin systeemiksi saadaan

$$\boldsymbol{y}'(t) = \boldsymbol{A}\boldsymbol{y}(t)\,, \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0,$$

missä

$$\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{I}_{n_1 n_2} \\ \boldsymbol{\Delta}_{h_1,h_2} & 0 \end{bmatrix} \qquad \text{ja} \qquad \boldsymbol{y}(t) = \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{w}(t) \end{bmatrix}\,,\quad \boldsymbol{y}_0 = \begin{bmatrix} \boldsymbol{f}_{h_1,h_2} \\ \boldsymbol{g}_{h_1,h_2} \end{bmatrix}.$$

Suoritetaan seuraavaksi aikadiskretointi. Olkoon diskretointiväli $\delta > 0$ ja merkitään $\boldsymbol{y}_j = \boldsymbol{y}(j\delta)$. Tällöin implisiittinen keskipistesääntö antaa yhtälöt

$$\boldsymbol{y}_{j+1} = \boldsymbol{y}_j + \delta\boldsymbol{A}\big(\tfrac{1}{2}(\boldsymbol{y}_j + \boldsymbol{y}_{j+1})\big)\,, \qquad j = 0,1,2,\ldots,$$

jotka voidaan kirjoittaa muotoon

$$(\boldsymbol{I} - \tfrac{\delta}{2}\boldsymbol{A})\boldsymbol{y}_{j+1} = (\boldsymbol{I} + \tfrac{\delta}{2}\boldsymbol{A})\boldsymbol{y}_j\,, \qquad j = 0,1,2,\ldots. \tag{5.25}$$

Tässä $\boldsymbol{y}_{j+1}$ saadaan siis edellisen $\boldsymbol{y}_j$ avulla ratkaisemalla lineaarinen yhtälöryhmä.

Kokeessa laskettiin $\boldsymbol{y}_1$ GMRESillä käyttäen sekä tavallista matriisien ILU(0)-pohjustusta että $\mathbb{R}$-lineaarista ILU(0)-pohjustusta. Lisäksi laskettiin kohdan 3.10 mukaisilla reaalilineaarisilla menetelmillä, jolloin edellä olevan yhtälön matriisi muunnettiin $\mathbb{R}$-operaattoriksi käyttäen tapaa (1.3). Diskretointiparametrit olivat $n_1 = n_2 = 32$ ja $\delta = 0.1$. Alkuehtona oli

$$f(\boldsymbol{x}) = \begin{cases} \frac{1}{2}\big(1 + \cos(\pi \|\boldsymbol{x} - [0.5\ 0.5]^T\|/0.1)\big), & \text{kun } \|\boldsymbol{x} - [0.5\ 0.5]^T\| < 0.1, \\ 0, & \text{muutoin}, \end{cases}$$

$$g(\boldsymbol{x}) = \begin{cases} 2 - 2\|x - [1\ 1]^T\|/0.3, & \text{kun } \|\boldsymbol{x} - [1\ 1]^T\| < 0.3, \\ 0, & \text{muutoin}. \end{cases}$$

Matriisien ILU(0)-pohjustus laskettiin käyttäen liitteen B funktiota `ilu_v2`. $\mathbb{R}$-lineaarisen pohjustuksen tapauksessa kerroinmatriisista $(\boldsymbol{I} - \frac{\delta}{2}\boldsymbol{A})$ tehtiin ensin $\mathbb{R}$-lineaarinen operaattori $\boldsymbol{\mathcal{M}} : \mathbb{C}^{n_1 n_2} \to \mathbb{C}^{n_1 n_2}$ kohdan (1.3) mukaan. Tämän jälkeen laskettiin $\boldsymbol{\mathcal{M}}$:lle $\mathbb{R}$-lineaarinen ILU(0)-hajotelma käyttäen funktiota `rl_ilu` ja tätä hajotelmaa käytettiin pohjustimena `MATLAB`in gmres-funktiolle. Kuvassa 5.7 näkyy ratkaisun jäännös piirrettynä iteraatioiden lukumäärää vasten. Nähdään, että $\mathbb{R}$-ILU(0) parantaa selkeästi suppenemisnopeutta verrattuna sekä pohjustamattomaan iterointiin että matriisien ILU(0):aan nähden. Algoritmi 3.10.1 ei kuitenkaan suppene $\mathbb{R}$-ILU(0):n kanssa. Nähdään myös, että tässä tehtävässä kohdan 3.10 reaalilineaariset iteratiiviset menetelmät eivät nopeuta suppenemista.

Diskretoinnista $n_1 = n_2 = 4$, saadaan kuva 5.8. Algoritmin 3.10.1 tapauksessa iteraatio ei suppene. Sen rivillä 3 laskettava vektori $\boldsymbol{v}$ on (numeerisesti lähes) nolla ensimmäisten iteraatioiden jälkeen, joten tässä tapauksessa algoritmi ei toimi.

5.4 Epälineaarinen Schrödingerin yhtälö

Tarkastellaan seuraavaa epälineaarista Schrödingerin yhtälöä

$$i\psi_t(\boldsymbol{x}, t) = \Delta\psi(\boldsymbol{x}, t) + |\psi(\boldsymbol{x}, t)|^2\psi(\boldsymbol{x}, t).$$

Rajoitutaan tässä myös neliöön $[0, 1] \times [0, 1] \subset \mathbb{R}^2$. Merkitään

$$\psi_{(i,j)}(t) = \psi(ih_1, jh_2), \qquad 0 \le i \le n_1 + 1,\ 0 \le j \le n_2 + 1.$$

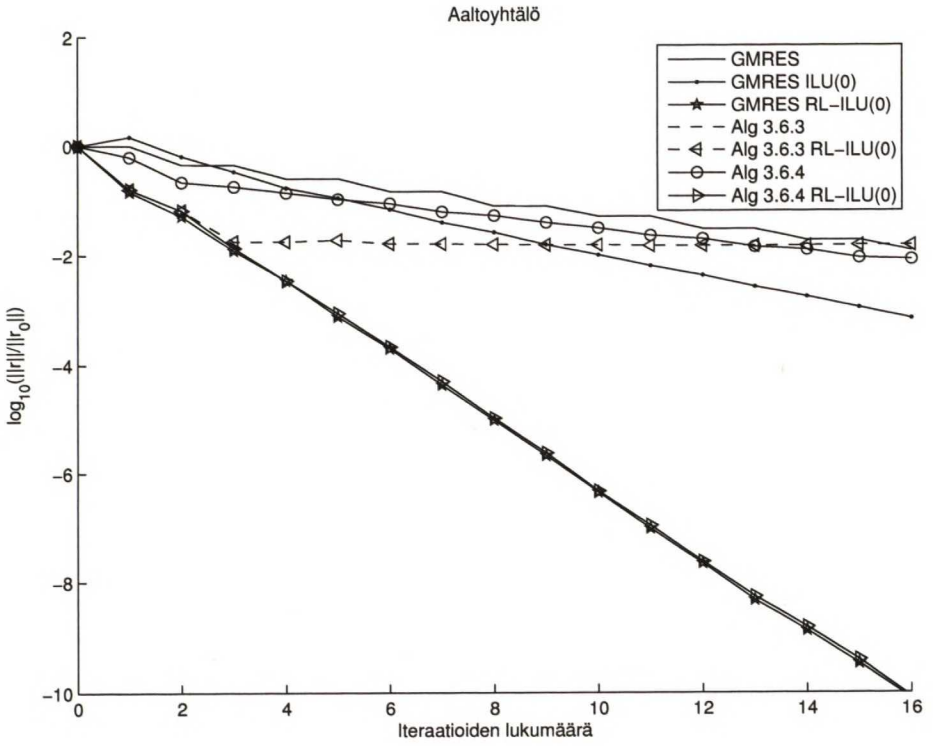
Olkoon tässäkin $\psi_{(0,j)}(t) = \psi_{(n_1+1,j)}(t) = \psi_{(i,0)}(t) = \psi_{(i,n_2+1)}(t) = 0$. Merkitään edelleen $w_{i+(j-1)n_1}(t) = \psi_{(i,j)}(t)$, $1 \le i \le n_1$, $1 \le j \le n_2$. Diskretoitu yhtälö on nyt

$$i\boldsymbol{w}'(t) = \boldsymbol{\Delta}_{h_1,h_2}\boldsymbol{w}(t) + \boldsymbol{G}(\boldsymbol{w}(t)), \tag{5.26}$$

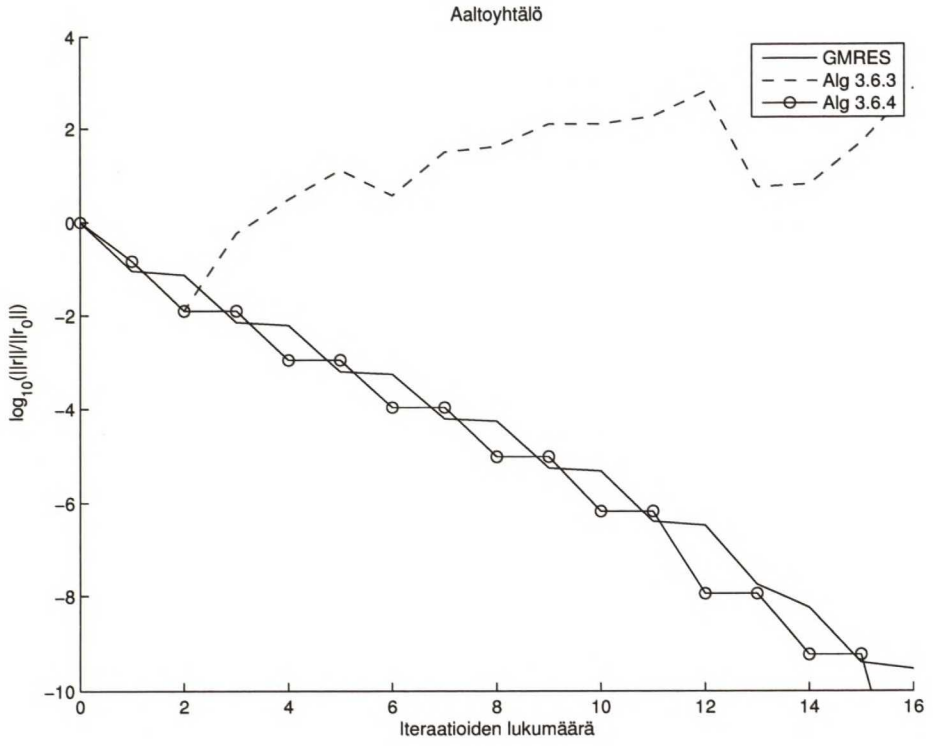
missä $\boldsymbol{G} : \mathbb{C}^{n_1 n_2} \to \mathbb{C}^{n_1 n_2}$, $\boldsymbol{G}(\boldsymbol{w})_i = |w_i|^2 w_i$.

Tehdään seuraavaksi aikadiskretointi askelpituudella $\delta > 0$. Käytetään differentiaaliyhtälölle $\boldsymbol{w}'(t) = \boldsymbol{\mathcal{M}}(\boldsymbol{w}(t))$, missä $\boldsymbol{\mathcal{M}} : \mathbb{C}^n \to \mathbb{C}^n$ on annettu ja $\boldsymbol{w}$ tuntematon, menetelmää

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \delta\boldsymbol{\mathcal{L}}(\boldsymbol{w}^k)\left(\tfrac{1}{2}(\boldsymbol{w}^k + \boldsymbol{w}^{k+1})\right), \qquad k = 0, 1, 2, \ldots, \tag{5.27}$$



Kuva 5.7: Aaltoyhtälön ratkaisun suhteellinen jäännös (tapaus $n_1 = n_2 = 32$). RL-ILU(0) on reaalilineaarinen ILU(0). Alg 3.10.1 ja Alg 3.10.2 viivat ovat päällekkäin, samoin GMRES RL-ILU(0) ja Alg 3.10.2 RL-ILU(0).



Kuva 5.8: Aaltoyhtälön ratkaisun suhteellinen jäännös (tapaus $n_1 = n_2 = 4$).

missä $\mathcal{L}(\boldsymbol{w}^k)$ on $\mathcal{M}$:n $\mathbb{R}$-linearisointi pisteessä $\boldsymbol{w}^k$. Tällöin siis $\mathcal{L}(\boldsymbol{w}^k)$ on vakion ja $\mathbb{R}$-lineaarisen operaattorin summa ja

$$\mathcal{M}(\boldsymbol{w}) = \mathcal{L}(\boldsymbol{w}^k)(\boldsymbol{w}) + o(|\boldsymbol{w} - \boldsymbol{w}^k|).$$

Differentiaaliyhtälön (5.26) tapauksessa on $\mathcal{M}(\boldsymbol{w}) = -i\boldsymbol{\Delta}_{h_1,h_2}\boldsymbol{w} - i\boldsymbol{G}(\boldsymbol{w})$ ja, koska

$$|v+u|^2(v+u) = (v+u)^2(\overline{v}+\overline{u}) = |v|^2v + 2|v|^2u + v^2\overline{u} + o(|u|),$$

sen $\mathbb{R}$-linearisointi pisteessä $\boldsymbol{w}^k$ on

$$\mathcal{L}(\boldsymbol{w}^k)(\boldsymbol{w}) = -i\Big(\boldsymbol{\Delta}_{h_1,h_2}\boldsymbol{w}^k + \boldsymbol{G}(\boldsymbol{w}^k) + (\boldsymbol{\Delta}_{h_1,h_2} + \boldsymbol{B}(\boldsymbol{w}^k))(\boldsymbol{w}-\boldsymbol{w}^k) + \boldsymbol{C}(\boldsymbol{w}^k)(\overline{\boldsymbol{w}-\boldsymbol{w}^k})\Big),$$

missä $\boldsymbol{B}(\boldsymbol{w})_{i,i} = 2|w_i|^2$, $\boldsymbol{C}(\boldsymbol{w})_{i,i} = w_i^2$ ja $\boldsymbol{B}(\boldsymbol{w}) \in \mathbb{C}^{n_1n_2\times n_1n_2}$:n ja $\boldsymbol{C}(\boldsymbol{w}) \in \mathbb{C}^{n_1n_2\times n_1n_2}$:n muut alkiot nollia. Sijoittamalla tämä menetelmään (5.27) saadaan

$$\begin{aligned}&\Big(i\boldsymbol{I} - \tfrac{\delta}{2}(\boldsymbol{\Delta}_{h_1,h_2} + \boldsymbol{B}(\boldsymbol{w}^k))\Big)(\boldsymbol{w}^{k+1} - \boldsymbol{w}^k) - \tfrac{\delta}{2}\boldsymbol{C}(\boldsymbol{w}^k)(\overline{\boldsymbol{w}^{k+1} - \boldsymbol{w}^k})\\ &= \delta\boldsymbol{\Delta}_{h_1,h_2}\boldsymbol{w}^k + \delta\boldsymbol{G}(\boldsymbol{w}^k).\end{aligned}$$

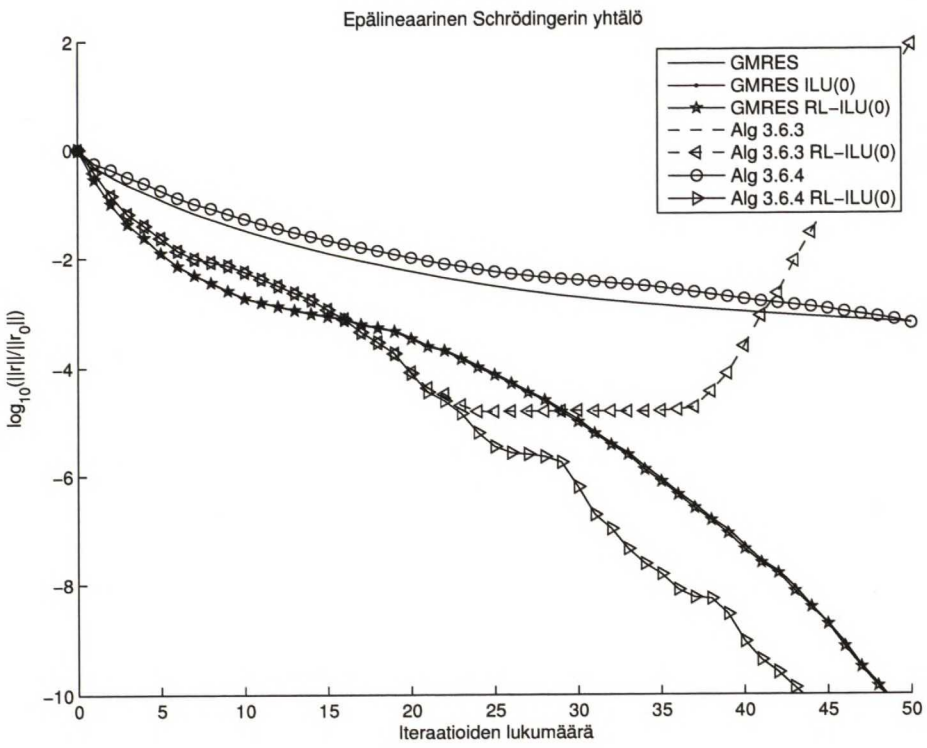
Reaalilineaariseksi yhtälöksi saadaan näin $\boldsymbol{M}\boldsymbol{z} + \boldsymbol{M}_{\#}\overline{\boldsymbol{z}} = \boldsymbol{b}$, missä

$$\begin{aligned}\boldsymbol{M} &= i\boldsymbol{I} - \tfrac{\delta}{2}(\boldsymbol{\Delta}_{h_1,h_2} + \boldsymbol{B}(\boldsymbol{w}^k)), \qquad \boldsymbol{M}_{\#} = -\tfrac{\delta}{2}\boldsymbol{C}(\boldsymbol{w}^k),\\ \boldsymbol{b} &= \delta\boldsymbol{\Delta}_{h_1,h_2}\boldsymbol{w}^k + \delta\boldsymbol{G}(\boldsymbol{w}^k),\end{aligned}$$

ja merkitään $\boldsymbol{z} = \boldsymbol{w}^{k+1} - \boldsymbol{w}^k$.

Kun alkuehto $\psi(\boldsymbol{x}, 0)$ on annettu, on se diskretoituna $(\boldsymbol{w}^0)_{i+(j-1)n_1} = \psi(ih_1, jh_2, 0)$. Numeerisessa kokeessa suoritettiin vastaavat laskut kuin edellisen kohdan aaltoyhtälön tapauksessa. Edellä olevasta reaalilineaarisesta yhtälöstä ratkaistiin $\boldsymbol{w}^1$ annetun alkuehtovektorin $\boldsymbol{w}^0$ perusteella. Parametrit olivat $n_1 = n_2 = 32$ ja $\delta = 0.1$ ja alkuehdoksi valittiin edellisen kohdan 5.3 alkuehtovektoreita käyttäen $\boldsymbol{w}^0 = \boldsymbol{f}_{h_1,h_2} + i\boldsymbol{g}_{h_1,h_2}$ (huom. i on tässä imaginaariyksikkö eikä indeksiluku). Matriisien GMRESia varten muunnettiin reaalilineaarinen yhtälö käyttäen tapaa (1.3).

Kuvassa 5.9 näkyy ratkaisun suhteellinen jäännös erilaisin menetelmin. Edelleen $\mathbb{R}$-ILU(0) tuo suppenemiseen parannusta, mutta nyt ILU(0) suppenee yhtä hyvin matriisien GMRESin kanssa. Lisäksi havaitaan, että algoritmia 3.10.1 vastaava menetelmä käyttäytyy edelleen huonosti.



Kuva 5.9: Epälineaarisen Schrödingerin yhtälön ratkaisun suhteellinen jäännös. Alg 3.10.1 ja Alg 3.10.2 viivat ovat päällekkäin, samoin GMRES ILU(0) ja RL-ILU(0) ovat päällekkäin.

Yhteenveto

Tässä työssä on jatkettu viitteen [1] aloittamaa reaalilineaaristen menetelmien tutkimusta. QR-hajotelman laskennassa tarvittava Householderin reaalilineaarinen muunnos tarkentui kohdassa 1.2 numeerisesti stabiilimpaan muotoon. Kohdissa 2.3 ja 2.3.1 ehdotettiin uutta tapaa tukea reaalilineaarista LU-hajotelmaa. Viitteen [1] GMRES-menetelmä esiteltiin kohdassa 3.8 ja Householderin muunnoksien tilalle vaihtoehdoksi tarjottiin uudet lauseen 3.8.1 esittämät reaalilineaariset Givens-rotaatiot. Luvun 5 numeeriset kokeet vahvistivat reaalilineaarisen GMRES-menetelmän nopeusedun reaalisten matriisien menetelmään verrattuna. Uutena kokeellisena havaintona kohdan 3.9 $\mathbb{C}$-linearisoitu yhtälö (3.54) on luvun 5 numeeristen kokeiden perusteella mielenkiintoinen mahdollisuus sähköisen impedanssitomografian tyyppisen $\overline{\partial}$-yhtälön ratkaisussa.

Toisaalta viitteessä [1] mainittu mahdollisuus reaalilineaaristen ILU-pohjustimien käytöstä reaalisten $2n \times 2n$-matriisien pohjustimina vaikuttaa hedelmättömältä. Reaalilineaarinen LU-hajotelma ja ILU-hajotelmat ovat tällöin erikoistapauksia yleisemmistä matriisien lohkohajotelmista. Tällaiset hajotelmat ovat yleisesti, riippuen sovelluskohteesta, toimiviksi tunnettuja. Tästä osoituksena myös luvun 5 kokeet vahvistavat $\mathbb{R}$-ILU(0) hajotelman pienentävän iteraatiokierroksia.

Kirjallisuutta

[1] T. Eirola, M. Huhtanen, J. von Pfaler. *Solution methods for $\mathbb{R}$-linear problems in $\mathbb{C}^n$.* SIAM J. Matrix Anal. Appl. 25 (2004), pp. 804-828.

[2] Marko Huhtanen, Olavi Nevanlinna. *Real linear matrix analysis.* Banach Center Publ. 75 (2007), pp. 171-189.

[3] J.L. Mueller, S. Siltanen. *Direct reconstruction of conductivities from boundary measurements.* SIAM J. Sci. Comput. 24 (2003), pp. 1232-1266.

[4] J.L. Mueller, S. Siltanen, D. Isaacson. *An implementation of the reconstruction algorithm of A. Nachman for the 2D inverse conductivity problem.* Inverse Problems 16 (2000), pp. 681-699.

[5] K. Knudsen, J.L. Mueller, S. Siltanen. *Numerical solution method for the d-bar equation in the plane.* Journal of Computational Physics 198 (2004), pp. 500-517.

[6] G. Vainikko. *Fast solvers of the Lippmann-Schwinger equation.* Direct and Inverse Problems of Mathematical Physics, Int. Soc. Anal. Appl. Comput., 5, Kluwer Acad. Publ., Dordrecht, 2000, pp. 423-440.

[7] Kari Astala, Lassi Päivärinta. *Calderón's inverse conductivity problem in the plane.* Annals of Mathematics 163 (2006), pp. 265-299.

[8] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, USA, 2003. ISBN 0-89871-534-2.

[9] Timo Eirola. *Krylov integrators for Hamiltonian systems.* Workshop on Exponential Integrators, 20-23.10.2004, Innsbruck, Itävalta. `http://techmath.uibk.ac.at/numbau/alex/events/files04/slides/timo.pdf`.

[10] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in C: the art of scientific computing.* 2nd Edition, Cambridge University Press, New York, USA, 1992. ISBN 0-521-43108-5.

[11] Roger A. Horn, Charles R. Johnson. *Matrix Analysis.* Cambridge University Press, 1987. ISBN 0-521-30586-1.

[12] Carl B. Boyer. *Tieteiden kuningatar.* Toinen painos, 2 osaa, Art House, 1995. ISBN 951-884-150-0. Englanninkielinen alkuperäisteos Carl B. Boyer, *A history of Mathematics*, 2nd Edition, Revised by Uta C. Merzbach, suomentanut Kimmo Pietiläinen.

[13] Carl D. Meyer. *History of Gaussian Elimination.* The 1982 Mathematical Calendar, Rome Press. `http://meyer.math.ncsu.edu/Meyer/PS_Files/GaussianEliminationHistory.pdf`.

[14] Henk A. van der Vorst. *Krylov subspace iteration.* Computing in Science and Engineering, vol. 2, no. 1 (2000), pp. 32-37.

[15] James H. Wilkinson. *Some Comments from a Numerical Analyst.* Journal of the ACM, vol. 18, no. 2 (1971), pp. 137-147.

[16] N. N. Lebedev. *Special Functions and Their Applications.* Dover, New York, 1972. Revised English Edition Translated and Edited by Richard A. Silverman.

Liite A

Matriisit

Matriisi $\boldsymbol{A}$ on $m \times n$-kokoinen kompleksilukualkioista a_{ij} koostuva taulukko. Riviindeksi i saa arvot $1, 2, \ldots, m$ ja sarakeindeksi j arvot $1, 2, \ldots, n$. Kaikkien $m \times n$-matriisien joukkoa merkitään $\mathbb{C}^{m \times n}$. Reaalisten matriisien joukkoa merkitään $\mathbb{R}^{m \times n}$.

Indeksien $i = j$ paikkoja sanotaan (pää)diagonaaliksi (l. lävistäjäksi). Matriisin transpoosi $\boldsymbol{A}^T$ saadaan kirjoittamalla rivit sarakkeiksi, tarkemmin $(\boldsymbol{A}^T)_{ij} = a_{ji}$. Matriisin hermitoinnissa otetaan lisäksi kompleksikonjugaatti $\boldsymbol{A}^* = \overline{\boldsymbol{A}}^T$.

Matriisi $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ on

(1) diagonaalimatriisi (l. lävistäjämatriisi), jos se on neliömatriisi ja $a_{ij} = 0$ kaikilla $i \neq j$,
(2) yläkolmiomatriisi, jos $a_{ij} = 0$ kaikilla $i > j$,
(3) alakolmiomatriisi, jos $a_{ij} = 0$ kaikilla $i < j$,
(4) Hessenbergin matriisi, jos $a_{ij} = 0$ kaikilla $i > j + 1$.

Matriisia $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ sanotaan neliömatriisiksi. Se on

(1) yksikkömatriisi, jos $a_{ii} = 1$ kaikilla i ja $a_{ij} = 0$ kaikilla $i \neq j$. Sitä merkitään $\boldsymbol{I}$ ja joskus tarkemmin $\boldsymbol{I}_n$.
(2) kääntyvä, jos on olemassa $\boldsymbol{B} \in \mathbb{C}^{n \times n}$ siten, että $\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}$. Tällöin merkitään $\boldsymbol{A}^{-1} = \boldsymbol{B}$.
(3) (vino)symmetrinen, jos $\boldsymbol{A}^T = (-)\boldsymbol{A}$.
(4) hermiittinen, jos $\boldsymbol{A}^* = \boldsymbol{A}$.
(5) normaali, jos $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^*$.
(6) unitaarinen, jos $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{I}$.
(7) ortogonaalinen, jos $\boldsymbol{A}$ on reaalinen ja $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$.

Avaruuden $\mathbb{C}^n$ vektorit esitetään yleensä $n \times 1$-(pysty)vektoreina. Luonnollinen kantavektori $\boldsymbol{e}_i$ on $n \times 1$-pystyvektori, jonka i:s alkio on 1 ja muut nollia. Vektoreiden sisätuloa merkitään $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^*\boldsymbol{x}$ ja vektorin normia $\|\boldsymbol{x}\| = (\boldsymbol{x}^*\boldsymbol{x})^{1/2}$. Vektorin $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T$ p-normi on $\|\boldsymbol{x}\|_p = (\sum_{k=1}^n |x_k|^p)^{1/p}$, kun $1 \leq p < \infty$, ja ∞-normi on $\|\boldsymbol{x}\|_\infty = \max_k |x_k|$.

Hermiittinen matriisi $\boldsymbol{A}$ on positiividefiniitti, jos $\langle \boldsymbol{Ax}, \boldsymbol{x} \rangle > 0$ kaikilla $\boldsymbol{x} \neq 0$. Luku $\lambda \in \mathbb{C}$ on neliömatriisin $\boldsymbol{A}$ ominaisarvo, jos on olemassa (ominaisvektori) $\boldsymbol{x} \neq 0$ siten,

että $\boldsymbol{Ax} = \lambda \boldsymbol{x}$. Kaikkien ominaisarvojen joukkoa kutsutaan $\boldsymbol{A}$:n spektriksi ja sitä merkitään $\sigma(\boldsymbol{A})$.

Matriisille $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ voidaan käyttää lohkomerkintää

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{D} & \boldsymbol{E} \end{bmatrix},$$

missä $\boldsymbol{B} \in \mathbb{C}^{p \times q}, \boldsymbol{C} \in \mathbb{C}^{p \times s}, \boldsymbol{D} \in \mathbb{C}^{r \times q}$ ja $\boldsymbol{E} \in \mathbb{C}^{r \times s}$, $m = p + r$, $n = q + s$. Tällöin alkiot ovat

$$a_{ij} = \begin{cases} b_{ij}, & \text{kun } 1 \leq i \leq p \text{ ja } 1 \leq j \leq q, \\ c_{i,j-q}, & \text{kun } 1 \leq i \leq p \text{ ja } q < j \leq n, \\ d_{i-p,j}, & \text{kun } p < i \leq m \text{ ja } 1 \leq j \leq q, \\ e_{i-p,j-q}, & \text{kun } p < i \leq m \text{ ja } q < j \leq n. \end{cases}$$

Lohkomerkinnän erikoistapauksena on $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_k \end{bmatrix}$, missä $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, ..., $\boldsymbol{q}_k \in \mathbb{C}^n$ ovat pystyvektoreita. Tällöin $\boldsymbol{Q}$ on $n \times k$-matriisi, jonka sarakkeet koostuvat vektoreista $\boldsymbol{q}_j$. Matriisi $\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{q}_{k+1} \end{bmatrix}$ saadaan lisäämällä matriisin $\boldsymbol{Q}$ viimeiseksi sarakkeeksi uusi pystyvektori $\boldsymbol{q}_{k+1}$.

Matriisin $\boldsymbol{A} = (a_{ij})$ saraketta j vastaavaa vektoria merkitään $\boldsymbol{a}_{*j}$ ja riviä i vastaavaa $\boldsymbol{a}_{i*}$. Matriisi $\boldsymbol{A}_{r:s,t:u}$ koostuu $\boldsymbol{A}$:n alkioista a_{ij}, joille $r \leq i \leq s$ ja $t \leq j \leq u$. Vektori $\boldsymbol{v}_{r:s}$ koostuu vektorin $\boldsymbol{v}$ komponenteista i, joille $r \leq i \leq s$.

Matriisien $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ ja $\boldsymbol{B} \in \mathbb{C}^{p \times q}$ Kroneckerin tulo on $(mp) \times (nq)$-matriisi

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

Jos matriisien $\boldsymbol{A}_1$ ja $\boldsymbol{A}_2$ tulo voidaan laskea ja samoin matriisien $\boldsymbol{B}_1$ ja $\boldsymbol{B}_2$ tulo, niin $(\boldsymbol{A}_1 \otimes \boldsymbol{B}_1)(\boldsymbol{A}_2 \otimes \boldsymbol{B}_2) = \boldsymbol{A}_1\boldsymbol{A}_2 \otimes \boldsymbol{B}_1\boldsymbol{B}_2$. Myös $(\boldsymbol{A} \otimes \boldsymbol{B})^* = \boldsymbol{A}^* \otimes \boldsymbol{B}^*$.

Liite B

MATLAB-ohjelmalistaukset

B.1 Kohta 2.3.1

```
function [L,La,U,Ua,V,P] = rl_luvp(M,Ma)

% Laskee LU-hajotelman kääntyvälle reaalilineaariselle operaattorille
% käyttäen Gaussin eliminointia osittaistuennalla.
%
% Palauttaa:
%  L,La    - Alakolmiomatriiseja. Matriisin L lävistäjä koostuu
%            ykkösistä ja matriisin La lävistäjä nollista.
%  U,Ua    - Yläkolmiomatriiseja.
%  V       - Yläkolmiomatriisi, jonka lävistäjä koostuu ykkösistä ja
%            jokainen rivi sisältää korkeintaan yhden muun nollasta
%            poikkeavan alkion.
%  P       - Permutaatiomatriisi.
%
% Laskettu hajotelma toteuttaa: op(L,La)*op(U,Ua)=V*P*op(M,Ma), missä
% esim. op(M,Ma) viittaa reaalilineaariseen operaattoriin, jonka
% lineaarinen osa on M ja anti-lineaarinen osa on Ma.
%

n=size(M,1); L=eye(n); La=zeros(n); U=M; Ua=Ma; V=eye(n); rperm=1:n;

for k=2:n

  % Etsitään nykyisestä sarakkeesta suurimman normin operaattori.

  ls=real(U(k-1:n,k-1)).^2+imag(U(k-1:n,k-1)).^2;
  ls=ls-real(Ua(k-1:n,k-1)).^2-imag(Ua(k-1:n,k-1)).^2;

  [sk,si]=max(abs(ls));
  si=si+k-2; sj=0;

  % Etsitään nykyisestä sarakkeesta pari skalaarioperaattoreita,
```

```
% joista voidaan muodostaa kääntyvä operaattori. Yritetään myös
% valita tämä pari siten, että niistä muodostetulla operaattorilla
% on iso normi parantaen yllä etsittyä.

for i=k-1:n-1
  for j=i+1:n
    qt=U(i,k-1)*conj(U(j,k-1))-Ua(i,k-1)*conj(Ua(j,k-1));
    qt2=conj(qt);
    qst=abs(qt)^2;
    skt=ls(i-k+2)+qst*ls(j-k+2);

    if (skt > 0)
      skt=skt+2*qst;
    else
      skt=2*qst-skt;
      qt=-qt;
    end

    if (skt > sk)
      sk=skt; si=i; sj=j; q=qt;
    end

    skt=ls(j-k+2)+qst*ls(i-k+2);

    if (skt > 0)
      skt=skt+2*qst;
    else
      skt=2*qst-skt;
      qt2=-qt2;
    end

    if (skt > sk)
      sk=skt; si=j; sj=i; q=qt2;
    end
  end
end

% Permutoidaan rivejä tarvittaessa. V:lle permutoidaan sarakkeita.

if (si ~= k-1)
  r=U(k-1,1:n); U(k-1,1:n)=U(si,1:n); U(si,1:n)=r;
  r=Ua(k-1,1:n); Ua(k-1,1:n)=Ua(si,1:n); Ua(si,1:n)=r;

  if (k > 2)
    r=L(k-1,1:k-2); L(k-1,1:k-2)=L(si,1:k-2); L(si,1:k-2)=r;
    r=La(k-1,1:k-2); La(k-1,1:k-2)=La(si,1:k-2); La(si,1:k-2)=r;
    r=V(1:k-2,k-1); V(1:k-2,k-1)=V(1:k-2,si); V(1:k-2,si)=r;
  end

  rs=rperm(k-1); rperm(k-1)=rperm(si); rperm(si)=rs;
```

```
    if (sj == k-1)
      sj = si;
    end
  end

  % Lisätään alapuolinen rivi (kerrottuna sopivalla kompleksiluvulla)
  % nykyiseen riviin, jolloin saadaan kääntyvä tukialkio.

  if (sj ~= 0)
    V(k-1,sj)=q;
    U(k-1,k-1:n)=U(k-1,k-1:n)+q*U(sj,k-1:n);
    Ua(k-1,k-1:n)=Ua(k-1,k-1:n)+q*Ua(sj,k-1:n);
    if (k > 2)
      L(k-1,1:k-2)=L(k-1,1:k-2)+q*L(sj,1:k-2);
      La(k-1,1:k-2)=La(k-1,1:k-2)+q*La(sj,1:k-2);
    end
  end

  a=U(k-1,k-1); b=Ua(k-1,k-1);
  w=[U(k:n,k-1),Ua(k:n,k-1)]/[a,b;b',a'];
  L(k:n,k-1)=w(:,1); La(k:n,k-1)=w(:,2);
  U(k:n,k:n)=U(k:n,k:n)-w*[U(k-1,k:n);conj(Ua(k-1,k:n))];
  Ua(k:n,k:n)=Ua(k:n,k:n)-w*[Ua(k-1,k:n);conj(U(k-1,k:n))];
  U(k:n,k-1)=zeros(n-k+1,1); Ua(k:n,k-1)=zeros(n-k+1,1);
end

% Luodaan permutaatiomatriisi permutaatioindeksivektorista.

P=zeros(n);

for k=1:n
  P(k,rperm(k))=1;
end
```

B.2 Kohta 4.2.1

Alla on MATLAB-esimerkkikoodi kohtaan 4.2.1. Tässä NZ on nollakuviomatriisi. Alkion arvo nolla tarkoittaa, että sitä vastaava indeksipari kuuluu nollakuvioon.

```
function [L, U] = ilu_v2(A, NZ)

    n = size(A, 1); L = eye(n); U = A;

    for i = 1:n
        for k = 1:i-1
            if NZ(i, k) == 0
                L(i, k) = 0;
```

```
            else
                L(i, k) = U(i, k) / U(k, k);
                U(i, k+1:n) = U(i, k+1:n) - L(i, k)*U(k, k+1:n);
            end
        end

        U(i, 1:i-1) = zeros(1, i-1);
        for j = i+1:n
            if NZ(i, j) == 0
                U(i, j) = 0;
            end
        end
    end
```

B.3 Kohta 4.2.2

Alla kohdan 4.2.2 funktio. Argumentti `droptol` on sama kuin em. kohdan ϵ.

```
function [L, U] = ilut(A, p, droptol)

    n = size(A, 1); L = eye(n); U = A;

    for i = 1:n
        rownorm = norm(A(i, :));

        for k = 1:i-1
            c = U(i, k) / U(k, k);
            if abs(c) < droptol*rownorm
                L(i, k) = 0;
            else
                L(i, k) = c;
                U(i, k+1:n) = U(i, k+1:n) - c*U(k, k+1:n);
            end
        end

        if i > 1
            U(i, 1:i-1) = zeros(1, i-1);
        end

        for j = i+1:n
            if abs(U(i, j)) < droptol*rownorm
                U(i, j) = 0;
            end
        end

        if i > p+1
            [s, si] = sort(abs(L(i, 1:i-1)));
            L(i, si(1:i-1-p)) = 0;
```

```
        end

        if i < n-p
            [s, si] = sort(abs(U(i, i+1:n)));
            U(i, i + si(1:n-i-p)) = 0;
        end
    end
```

B.4 Kohta 4.3.1

```
%
% Lasketaan reaalilineaarisen operaattorin z -> M*z + Ma*conj(z)
% ILU-hajotelma käyttäen nollakuviota NZ.
%

function [L, La, U, Ua] = rl_ilu(M, Ma, NZ)

    n = size(M, 1);
    L = speye(n); La = spalloc(n, n, 3*n);
    U = spalloc(n, n, 3*n); Ua = spalloc(n, n, 3*n);

    for i = 1:n
        w = full(M(i,:)); wa = full(Ma(i,:));

        for k = 1:i-1
            if NZ(i, k) == 0
                w(k) = 0; wa(k) = 0;
            else
                a = U(k, k); b = Ua(k, k);
                c = [w(k), wa(k)] / [a, b; b', a'];
                w(k) = c(1); wa(k) = c(2);
                w(k+1:n) = w(k+1:n) - ...
                    c*[U(k, k+1:n); conj(Ua(k, k+1:n))];
                wa(k+1:n) = wa(k+1:n) - ...
                    c*[Ua(k, k+1:n); conj(U(k, k+1:n))];
            end
        end

        for k = i+1:n
            if NZ(i, k) == 0
                w(k) = 0; wa(k) = 0;
            end
        end

        w = sparse(w); wa = sparse(wa);
        if i > 1
            L(i, 1:i-1) = w(1:i-1); La(i, 1:i-1) = wa(1:i-1);
        end
```

```
        U(i, i:n) = w(i:n); Ua(i, i:n) = wa(i:n);
    end
```

B.5 Algoritmi 4.3.2

Alla algoritmia 4.3.2 vastaava funktio. Tämä hyödyntää MATLABin harvojen matriisien käsittelyfunktioita.

```
function [L, La, U, Ua] = rl_ilut(M, Ma, p, droptol)

    n = size(M, 1);
    L = speye(n); La = spalloc(n, n, 3*n);
    U = spalloc(n, n, 3*n); Ua = spalloc(n, n, 3*n);

    for i = 1:n
        w = full(M(i, :)); wa = full(Ma(i, :));
        rownorm = sum(abs(w) + abs(wa));

        for k = 1:i-1
            a = U(k, k); b = Ua(k, k);
            c = [w(k), wa(k)] / [a, b; b', a'];
            if (abs(c(1)) + abs(c(2))) < droptol*rownorm
                w(k) = 0; wa(k) = 0;
            else
                w(k) = c(1); wa(k) = c(2);
                w(k+1:n) = w(k+1:n) - ...
                    c*[U(k, k+1:n); conj(Ua(k, k+1:n))];
                wa(k+1:n) = wa(k+1:n) - ...
                    c*[Ua(k, k+1:n); conj(U(k, k+1:n))];
            end
        end

        for k = i:n
            if (abs(w(k)) + abs(wa(k))) < droptol*rownorm
                w(k) = 0; wa(k) = 0;
            end
        end

        if i > p+1
            [s, si] = sort(abs(w(1:i-1)) + abs(wa(1:i-1)));
            L(i, si(i-p:i-1)) = w(si(i-p:i-1));
            La(i, si(i-p:i-1)) = wa(si(i-p:i-1));
        else
            if (i > 1)
                L(i, 1:i-1) = w(1:i-1); La(i, 1:i-1) = wa(1:i-1);
            end
        end
```

```
    if i < n-p
        [s, si] = sort(abs(w(i+1:n)) + abs(wa(i+1:n)));
        U(i, i) = w(i); Ua(i, i) = wa(i);
        U(i, i + si(n-i-p+1:n-i)) = w(i + si(n-i-p+1:n-i));
        Ua(i, i + si(n-i-p+1:n-i)) = wa(i + si(n-i-p+1:n-i));
    else
        U(i, i:n) = w(i:n); Ua(i, i:n) = wa(i:n);
    end
end
```