

Susanne Jauhiainen

Poissonin yhtälön nopeat ratkaisijat

Tietotekniikan kandidaatintutkielma

27. huhtikuuta 2016

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Susanne Jauhiainen

Yhteystiedot: `sumahein@student.jyu.fi`

Työn nimi: Poissonin yhtälön nopeat ratkaisijat

Title in English: Fast Poisson solvers

Työ: Kandidaatintutkielma

Sivumäärä: 22+0

Tiivistelmä: Tutkielmassa esitellään Poissonin yhtälö sekä sen diskretointi. Lisäksi käydään läpi kaksi nopeaa numeerista menetelmää yhtälön ratkaisemiseksi. Yksinkertaisuuden vuoksi rajoitutaan kaksiulotteisiin tehtäviin, joissa on voimassa Dirichlet'n reunaehto. Ensimmäinen menetelmä on monihilamenetelmä, joka on iteraatiivinen menetelmä, ja toisena syklinen reduktio, joka on suora menetelmä. Molemmat menetelmät ovat hyvin tehokkaita sekä helposti rinnakkaistuvia.

Avainsanat: Poissonin yhtälö, monihilamenetelmä, syklinen reduktio

Abstract: In this thesis we introduce Poisson's equation and its discretization. In addition we go through two fast numerical methods for solving the equation. The thesis is limited only to two-dimensional cases with Dirichlet boundary condition. The first method is the multigrid method which is an iterative method and the second method is the cyclic reduction which is a direct method. Both methods are very efficient and parallel.

Keywords: Poisson equation, multigrid, cyclic reduction

Sisältö

| | | |
|---|---|----|
| 1 | JOHDANTO | 1 |
| 2 | POISSONIN YHTÄLÖ | 3 |
| | 2.1 Poissonin reuna-arvot tehtävä | 3 |
| | 2.2 Poissonin yhtälön diskretoitu muoto | 3 |
| | 2.3 Häiriöalttius | 5 |
| 3 | MONIHILAMENETELMÄ | 6 |
| | 3.1 Monihilamenetelmän peruseriaate | 6 |
| | 3.2 Rajoittaminen ja prolungaatio | 8 |
| | 3.3 Tasoittaminen | 9 |
| | 3.4 Algebrallinen monihilamenetelmä | 10 |
| | 3.5 Graafipohjainen monihilamenetelmä | 11 |
| | 3.6 Menetelmän analysointia | 11 |
| 4 | SYKLINEN REDUKTIO | 13 |
| 5 | MENETELMIEN VERTAILUA | 15 |
| 6 | YHTEENVETO | 16 |
| | LÄHTEET | 17 |

1 Johdanto

Poissonin yhtälö kuuluu elliptisten osittaisdifferentiaaliyhtälöiden ryhmään. Sen avulla voidaan mallintaa esimerkiksi sähköstatiikkaa, konetekniikkaa ja teoreettista fysiikkaa. Sovelluskohteesta riippuen yhtälölle voidaan asettaa erilaisia reunaehtoja, joiden avulla muodostetaan Poissonin reuna-arvotehtäviä. Tälle tehtävälle on harvoin mahdollista laskea tarkkaa ratkaisua, jonka vuoksi sille halutaan numeerisesti laskea approksimaatio. Jotta numeerinen ratkaiseminen tietokoneella olisi mahdollista, täytyy Poissonin yhtälö ensin diskretoida eli jakaa tietokoneen ymmärtämiin osiin. Luvussa 2 esitellään Poissonin yhtälö ja sen diskretointi viiden pisteen menetelmällä. Saatavan diskretoidun muodon ratkaisemista varten on kehitetty useita eri menetelmiä, joista tehokkaita ovat esimerkiksi nopea Fourier muunnosmenetelmä, syklinen reduktio -menetelmä, FACR-menetelmä, PSCR-menetelmä (Rossi, Heikkola ja Toivanen 1998) sekä monihilamenetelmä. Tässä tutkielmassa esitellään monihilamenetelmä sekä syklinen reduktio. Syklinen reduktio -menetelmä käydään läpi melko lyhyesti ja monihilamenetelmään keskitytään yksityiskohtaisemmin, koska se tiedetään erittäin tehokkaaksi sekä tieteellisen tutkimuksen kannalta mielenkiintoiseksi.

Monihilamenetelmä on yksi tehokkaimmista iteratiivisista menetelmistä elliptisille yhtälöille. Se on ainut tunnettu menetelmä, jossa iteraatioiden määrä voidaan pitää vakiona riippumatta yhtälöryhmän koosta (Trottenberg, Oosterlee ja Schüller 2001). Monihilamenetelmää on viime aikaisessa tutkimuksessa hyödynnetty muun muassa vaatteiden simulointiin (Jeon ym. 2013), kuvankäsittelyyn (Stümer, Köstler ja Rüdde 2007), seismiseen tomografiaan (Kamath ym. 2015) sekä meriveden lämpötilan ennustamiseen (Wei ym. 2008). Menetelmä perustuu siloittamiseen ja ratkaisua approksimoivan korjauksen laskemiseen vain vähän laskentaresursseja vaativalla harvalla diskretointihilalla. Monihilamenetelmän käyttöön on useita erilaisia strategioita, jotka perustuvat harvempien ja tiheämpien hilojen sopivaan vuorotteluun. Menetelmä käydään läpi luvussa 3. Monihilamenetelmät voidaan jakaa geometriisiin ja algebrallisiin, joista tässä tutkielmassa algebrallinen esitellään luvussa 3.4 se-

kä siihen pohjautuva graafipohjainen monihilamenetelmä luvussa 3.5.

Syklinen reduktio on suora menetelmä Poissonin diskretoidun muodon ratkaisemiseksi suorakulmaisessa alueessa. Se on hyvin tunnettu menetelmä tridiagonaalisten yhtälöryhmien ratkaisemiseksi. Viimeaikaisessa tutkimuksessa syklinen reduktio -menetelmää on hyödynnetty esimerkiksi ohjelmoitaviin näytönohjaimiin (Myllykoski 2011) sekä kokoonpuristuviin viskooseihin virtasimulaatioihin (Esfahanian ym. 2008). Menetelmä perustuu parittomien rivien eliminointiin, kunnes jäljellä on vain yksi rivi. Tämän jälkeen palataan askeleet takaisinpäin ratkaisten kunkin vaiheen parittomat rivit. Luvussa 4 esitellään syklinen reduktio -menetelmän perusperiaate.

Myllykoski on opinnäytetöissään (2010, 2011) esitellyt Poissonin yhtälön sekä sille eri ratkaisumenetelmiä, esimerkiksi Fourier-muunnosmenetelmän sekä syklisen reduktion. Tässä tutkielmassa on esitetty syklinen reduktio -menetelmästä versio, joka hyödyntää laskemiseen matriisin polynomien juuria ja käyttää tätä kautta vähemmän tietokoneen muistia. Lisäksi syklinen reduktio -menetelmää on verrattu tässä tutkielmassa monihilamenetelmään, koska se on erittäin tehokas, kiinnostava ja ajankohtainen menetelmä. Luvussa 5 menetelmiä vertaillaan niiden aikavaativuuden, rinnakkaistettavuuden sekä muistin tarpeen suhteen. Molemmat menetelmät todetaan tehokkaiksi sekä hyvin rinnakkaistuviksi. Viimeinen luku sisältää yhteenvedon, jossa perustellaan mikä menetelmä kannattaa valita tietyssä sovellus-tilanteessa.

2 Poissonin yhtälö

Osittaisdifferentiaaliyhtälöä

$$\nabla^2 u = f \quad (2.1)$$

kutsutaan Poissonin yhtälöksi. Oikean puolen funktio f on voimatermi ja u on ratkaistava funktio. Derivaattaoperaattori nabla, ∇ , tarkoittaa gradienttia, esimerkiksi kaksiulotteisessa avaruudessa sen toinen potenssi, ∇^2 , on

$$\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}. \quad (2.2)$$

Tätä toista potenssia kutsutaan myös Laplace-operaattoriksi ja merkitään Δ . Kirjoittamalla tämä auki saadaan Poissonin yhtälö muotoon

$$\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right) u(x_1, x_2) = f(x_1, x_2). \quad (2.3)$$

2.1 Poissonin reuna-arvotehtävä

Poissonin yhtälölle voidaan tilanteesta riippuen asettaa erilaisia reunaehtoja. Yksinkertaisuuden vuoksi asetetaan yhtälölle Dirichlet'n reunaehto, $u = 0$, jolloin avaruudessa \mathbb{R}^n yhtälö saa muodon

$$\nabla^2 u = f, \quad \forall x \in \Omega, \quad (2.4)$$

$$u = g, \quad \forall x \in \partial\Omega, \quad (2.5)$$

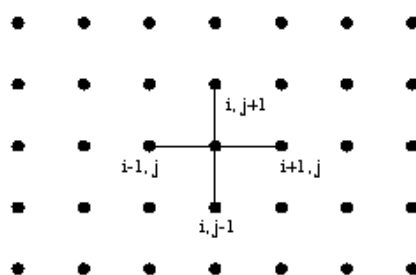
jossa Ω on avaruuden \mathbb{R}^n avoin osajoukko, $\partial\Omega$ sen reuna ja $f: \Omega \rightarrow \mathbb{R}$ ja $g: \partial\Omega \rightarrow \mathbb{R}$ jatkuvia funktioita. Tässä tutkielmassa rajoitutaan kaksiulotteisiin tapauksiin, joissa on voimassa Dirichlet'n reunaehto $u(x_1, x_2) = 0$, kun $(x_1, x_2) \in \partial\Omega$.

2.2 Poissonin yhtälön diskretoitu muoto

Poissonin tehtävälle on harvoin mahdollista laskea tarkkaa ratkaisua, joten sille täytyy laskea likimääräinen ratkaisu, eli approksimaatio, numeerisesti. Jotta yhtälö voitaisiin numeerisesti ratkaista, tulee se ensin diskretoida eli jakaa tehtävä ja alue

osiin. Näin se saadaan muunnettua tietokoneen ymmärtämään muotoon. Poissonin yhtälö voidaan diskretoida esimerkiksi differenssi-, elementti- tai tilavuusmenetelmällä. Näistä differenssi- ja tilavuusmenetelmiä käytetään perinteisesti säännöllisille suorakulmaisille alueille, kun taas elementtimenetelmää käytetään usein epäsäännöllisen alueen tapauksessa (Trottenberg, Oosterlee ja Schüller 2001).

Sutmann, Gibbort ja Lippert (2010) esittävät Poissonin tehtävän diskretoinnin viiden pisteen differenssimenetelmällä suorakulmion muotoisessa tasavälisessä hilassa. Tällöin laskenta-alueeseen muodostetaan hila eli verkko, jonka pisteissä ratkaisua tarkastellaan ja ratkaisu tietyssä verkon pisteessä riippuu solmun nelinaapurustosta. Verkon solmut numeroidaan x_i -suunnassa indeksillä i ja x_j -suunnassa indeksillä j (ks. kuvio 1).



Kuvio 2.1. Laskenta-alue jaettuna tasaväliseksi pisteistöksi. Kuvassa solmun (i,j) nelinaapurusto.

Sijoittamalla toisen asteen keskeisdifferenssiapproksimaatio

$$\left(\frac{\partial^2}{\partial x_i^2}\right) = \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2}, \quad i = 1, 2, \quad (2.6)$$

Poissonin yhtälöön (2.3) saadaan esitettyä se diskreetissä muodossa (Bolten 2010)

$$\frac{1}{h^2}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}) = f_{i,j}, \quad (2.7)$$

jossa $u_{i,j}$ ja $f_{i,j}$ ovat funktioiden u ja f arvot pisteessä (ih, jh) , h pisteiden välien pituus. Asetetaan yhtälölle lisäksi Dirichlet'n reunaehto $u = 0$, kun $(ih, jh) \in \partial\Omega$. Tällöin Poissonin yhtälö voidaan muuttaa matriisimuotoon

$$Au = f, \quad (2.8)$$

jossa

$$A = \begin{bmatrix} D & I & & & \\ I & D & I & & \\ & & \ddots & \ddots & \ddots \\ & & & I & D \end{bmatrix} \in \mathbb{R}^{N \times N}, u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N \text{ ja } f = h^2 \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \in \mathbb{R}^N, \quad (2.9)$$

$$D = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -4 \end{bmatrix} \in \mathbb{R}^{n \times n}, u_i = \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ \vdots \\ u_{i,n} \end{bmatrix}, \in \mathbb{R}^n, f_i = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,n} \end{bmatrix} \in \mathbb{R}^n, \quad (2.10)$$

jossa $N = n \times n$ ja I on $n \times n$ identtinen matriisi.

2.3 Häiriöalttius

Vaikka tehtävä $Au = f$ voidaan periaatteessa ratkaista tarkasti, niin käytännössä numeerinen ratkaisu on aina likiarvo. Parametreihin liittyy aina epätarkkuuksia ja liukulukuja käytettäessä aritmeettiset operaatiot eivät ole tarkkoja. Ratkaisun u suhteellista virhettä voidaan arvoida esimerkiksi yhtälöllä (Quarteroni, Sacco ja Saleri 2007)

$$\frac{\|e\|}{\|x\|} \leq \kappa(A) \frac{\|\delta f\|}{\|f\|}, \quad (2.11)$$

jossa $e = x - \hat{x}$ on syntyvä virhe, δf on f :n virhe ja lukua $\kappa(A) = \|A\| \|A^{-1}\|$ kutsutaan matriisin A häiriöalttiudeksi. Poissonin yhtälön tapauksessa saadaan häiriöalttius laskettua yhtälöstä

$$\kappa(A) = \left| \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \right|, \quad (2.12)$$

jossa $\lambda_{\max}(A)$ ja $\lambda_{\min}(A)$ ovat matriisin A suurin ja pienin ominaisarvo.

3 Monihilamenetelmä

Monihilamenelmä hyödyntää ratkaisun laskentaan sekä harvempia että tiheämpiä hilatasoja. Harvemmassa verkossa ratkaiseminen on nopeaa ja tiheämmällä verkolla saadaan tarkkoja ratkaisuja. Menetelmän ideana on käyttää harvoja ja tiheitä verkkoja sopivasti vuorotellen. Ratkaisu haetaan tiheimmällä tasolla ja ratkaisua harvemmillä tasoilla vastaa tiheämmällä tasolla laskettu residuaali.

3.1 Monihilamenetelmän peruseriaate

Esitellään monihilamenetelmän peruseriaa yhtälölle (2.8), jossa u on etsitty ratkaisu (Saraniti ym. 1996). Olkoon v^k arvio u :lle k :nnen iteraation jälkeen. Määritellään residuaali r^k eli arvion v^k poikkeama etsitystä arvosta u

$$r^k = f - Av^k. \quad (3.1)$$

Seuraavaksi määritellään korjaus e^k arviolle v^k

$$e^k = u - v^k. \quad (3.2)$$

Yhtälöistä (3.1) ja (3.2) saadaan

$$Ae^k = r^k. \quad (3.3)$$

Kun ratkaistaan korjaus e^k yhtälöstä (3.3), saadaan

$$v^{k+1} = v^k - e^k. \quad (3.4)$$

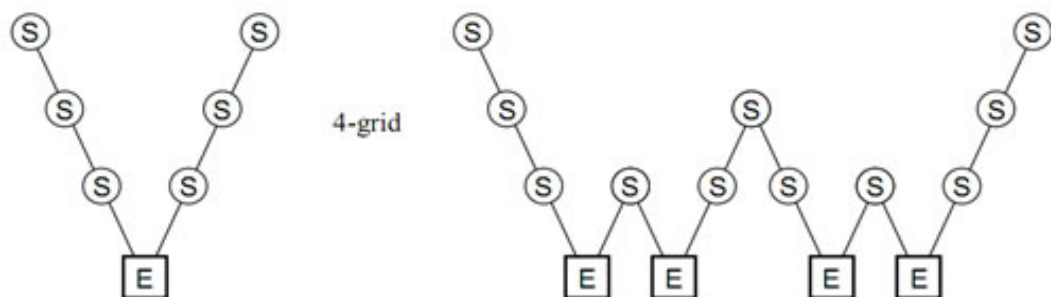
Nyt voidaan esittää yksinkertainen kahden tason monihila-algoritmi (Gholami ym. 2015, Siikonen 2014).

1. Tasoitetaan arvio v^k tasolla n .
2. Lasketaan residuaali yhtälöstä (3.1) ja siirretään se harvemmalle tasolle $n - 1$. Siirtoa kutsutaan rajoittamiseksi, katso luku 3.2.
3. Lasketaan korjaus e^k yhtälöstä (3.3) tasolla $n - 1$ ja siirretään se tiheämmälle tasolle n . Siirtoa kutsutaan prolongaatioksi, katso luku 3.2.

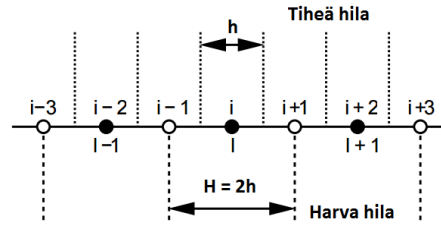
4. Lasketaan v^{k+1} yhtälöstä (3.4).
5. Tasoitetaan arvio v^{k+1} tasolla n .

Monihila-algoritmi voidaan laajentaa myös useammalle tasolle. Helpoin laajentamistapa on ajatella toisen tason ratkaisua uutena ratkaistavana ongelmana (Siikonen 2014). Sen sijaan, että saatu ratkaisu iteroitaisiin tarkaksi ja siirrettäisiin ensimmäiselle tasolle, voidaan soveltaa edellä esitettyä algoritmia toisen ja kolmannen tason välille ja niin edelleen.

Jokaisella tasolla voidaan tehdä yllä esitetystä kahden tason algoritmista γ iteraatioita. Jos valitaan esimerkiksi $\gamma = 1$, kutsutaan monihilasykliä V-sykliksi, kun taas valinnalla $\gamma = 2$ päädytään W-sykliin (Bolten 2010). Ajallisesti vaativinta monihilamenetelmässä on tasoittaminen. Tämän vuoksi tasoitusta tiheimmällä tasolla kannattaa tehdä mahdollisimman vähän. Esimerkiksi V-syklissä (ks. kuvio 2) siirrytään hilataso kerrallaan alaspäin ja sen jälkeen takaisin ylöspäin. Toinen yleinen vaihtoehto on W-sykli, jossa tasoja harvemmillä hiloilla vaihdellaan kuvion 2 mukaisesti. Näistä W-sykli on paras kompromissi iteraatoiden määrän sekä yhteen iteraatioon liittyvän laskenta-ajan välillä (Saraniti ym. 1996). W-sykli myös konvergoi nopeammin kuin V-sykli (Trottenberg, Oosterlee ja Schüller 2001).



Kuvio 3.1. Yleisimmät monihila rakenteet, V- ja W- syklit. Ympyrät S metkitsevät tasoittamista, E tarkkaa ratkaisua. Tässä esimerkissä hilatasoja on neljä, mutta periaatteessa niitä voi olla enemmän tai vähemmän.



Kuvio 3.2. Yksinkertainen esimerkki. (Siikonen, s. 144)

3.2 Rajoittaminen ja prolongaatio

Edellisen luvun algoritmissa haetaan approksimaatio korjaukselle e^k käyttäen harvempaa hilaa. Siikonen (2014) on luentomonisteessaan esittänyt yksinkertaisen esimerkin rajoittamisesta eli residuaalin siirtämisestä harvemmalle tasolle, yksiulotteisessa tapauksessa. Yksinkertaisin tapa hilan harventamiseen on jättää pois joka toinen solmukohta (ks. kuvio 3). Kirjoitetaan ensin yhtälö (3.3) esimerkin kannalta parempaan muotoon

$$\frac{1}{h^2}(e_{i+1}^k - 2e_i^k + e_{i-1}^k) = r_i^k. \quad (3.5)$$

Koska yksi harvan hilan väli peittää yhden tiheen hilan välin kokonaan ja sen viereiset välit puoleen väliin asti, voidaan ajatella yhtälöä muodossa

$$\frac{1}{(2h)^2}(e_{i+2}^k - 2e_i^k + e_{i-2}^k), \quad (3.6)$$

joka voidaan kaavan (3.5) avulla kirjoittaa muodossa

$$\frac{1}{4}(r_{i+1}^k - 2r_i^k + r_{i-1}^k). \quad (3.7)$$

Käyttäen harvan hilan merkintöjä voidaan kaavat (3.6) ja (3.7) kirjoittaa nyt muotoon

$$\frac{1}{H^2}(e_{l+1}^k - 2e_l^k + e_{l-1}^k) = r_l^n. \quad (3.8)$$

Tästä saadaan kaava residuaalin siirtoa eli rajoittamista varten:

$$r_l^n = \frac{1}{4}(r_{i+1}^k - 2r_i^k + r_{i-1}^k). \quad (3.9)$$

Matriisimuodossa esitettynä tämä on

$$\begin{bmatrix} 1/4 & -1/2 & 1/4 \end{bmatrix}. \quad (3.10)$$

Vastaava kaava saadaan laskettua samoin myös korkeammilla dimensioilla. Esimerkiksi kaksiulotteisessa saatava kaava matriisimuodossa esitettynä olisi (Saraniti ym. 1996)

$$\begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}. \quad (3.11)$$

Kun residuaali on siirretty harvemmälle hilalle, lasketaan korjaus e^k ja siirretään se takaisin tiheämmälle hilalle. Tätä siirto kutsutaan prolongaatioksi tai takaisin interpoloinniksi. Tälle vastaava kaava yksiulotteisessa tapauksessa olisi

$$e_i^k = \frac{1}{2}(e_{i+1}^k - 2e_i^k + e_{i-1}^k) \quad (3.12)$$

ja matriisimuodossa esitettynä

$$\begin{bmatrix} 1/2 \\ -1 \\ 1/2 \end{bmatrix} \quad (3.13)$$

ja esimerkiksi kaksiulotteisessa tapauksessa (Saraniti ym. 1996)

$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}. \quad (3.14)$$

3.3 Tasoittaminen

Tehokas monihilamenetelmä vaatii tasoittamista, jotta ero tarkan ratkaisun, u ja valitsevan approksimaation v^k välillä olisi tasainen. Tasoittamiseen voidaan käyttää esimerkiksi Gauss–Seidel- tai Jacobi-menetelmää, jotka ovat klassisia iteratiivisia ratkaisumenetelmiä (Trottenberg, Oosterlee ja Schüller 2001). Näistä menetelmistä Jacobi on helpommin rinnakkaistettavissa (Yousef 2000).

Bolten (2010) esittää Jacobi-menetelmän, jossa

$$u^{(k+1)} = u^{(k)} - D^{-1}(Au^{(k)} + b), \quad k = 1, 2, \dots, n, \quad (3.15)$$

jossa $u^{(0)}$ on alkuarvaus ja n ratkaisemiseen tarvittavien iteraatioiden lukumäärä. Jacobi-menetelmän konvergoimista voidaan nopeuttaa vaimentamalla (Trottenberg,

Oosterlee ja Schüller 2001). Tällöin valitaan sopiva ω ja saadaan kaava muotoon

$$u^{(k+1)} = u^{(k)} - \omega D^{-1}(Au^{(k)} + b), \quad k = 1, 2, \dots, n. \quad (3.16)$$

Gauss–Seidel-menetelmän etu Jacobiin nähden on, että se ei tyypillisesti tarvitse vaimentamista (Göddeke ja Strzodka 2011). Rinnakkaistaminen on kuitenkin monimutkaisempaa kuin Jacobi-menetelmällä.

3.4 Algebrallinen monihilamenetelmä

Monihilamenetelmät voidaan jakaa kahteen eri kategoriaan, geometrisiin ja algebrallisiin. Pääasiallinen ero näiden välillä on, että algebrallinen monihilaratkaisija ei tarvitse muuta tietoa kuin alkuperäisen yhtälöryhmän, kun taas geometrinen ratkaisija riippuu myös diskretoinnissa käytettävän verkon geometriasta (Gholami ym. 2015). Tässä luvussa esitellään algebrallinen monihilamenetelmä.

Algebrallisessa monihilamenetelmässä harvemmat tasot ja niiden operaattorit muodostetaan kerroinmatriisin avulla. Tämän vuoksi menetelmä on yksinkertainen, eikä tarvitse mitään geometrista tietoa, ja eri hilatasoilla tarvitsee muodostaa vain residuaalit sekä matriisi A . Olkoon n hilatasojen määrä, joilloin yhtälö (3.3) tiheimmällä tasolla n on siis

$$A_{\ell}e_{\ell} = r_{\ell}. \quad (3.17)$$

Harvemmillä tasoilla yhtälö tulee muotoon

$$A_{\ell-m}e_{\ell-m} = r_{\ell-m} \quad m = 1, \dots, \ell - 1. \quad (3.18)$$

Merkitään tarvittavaa rajoitusoperaattoria, eli kerroinmatriisia, R :llä ja muodostetaan harvemman tason matriisit $A_{\ell-m}$. Kerrotaan ensin rajoitusoperaattorilla R

$$RA_{\ell-m}e_{\ell-m} = Rr_{\ell-m} \quad m = 1, \dots, \ell - 1. \quad (3.19)$$

jonka jälkeen siirretään virhevektori harvemmalle tasolle prolongaatio-operaattorin P avulla. Yksinkertaisin valinta prolongaatio-operaattoriksi on $P = R^T$ (Siikonen 2014)

$$RA_{\ell-m}Pe_{\ell-m-1} = Rr_{\ell-m} \quad m = 1, \dots, \ell - 1. \quad (3.20)$$

Merkitään nyt $A_{\ell-m-1} = RA_{\ell-m}P$ ja $r_{\ell-m-1} = Rr_{m-\ell}$, saadaan harvemman tason yhtälö muodostettua.

$$A_{\ell-m}e_{\ell-m} = r_{\ell-m} \quad m = 2, \dots, \ell - 1. \quad (3.21)$$

3.5 Graafipohjainen monihilamenetelmä

Graafipohjainen monihilamenetelmä on algebrallisen monihilamenetelmän muunnos. Kickinger on esittänyt algebrallisen monihilamenetelmän, jossa harvempien tasojen muuttujat voidaan valita kerroinmatriisiin naapurisuusgraafin avulla. Pennanen (2010) on kehittänyt menetelmää ja esittänyt väitöskirjassaan graafipohjaisen monihilamenetelmän. Menetelmässä tärkein parannus on Dirichlet'n reunehtoien eliminointi yhtälöryhmästä vasta kun kaikki harvempien tasojen matriisit on muodostettu. Lisäksi harvemman tason muodostamiseen voidaan käyttää naapurisuusgraafin sijaan mitä tahansa tehtävään liittyvää graafia. Pennanen väitöskirjassa on valittu rajoitusmatriisiksi R , jonka komponentit ovat

$$R_{i,j} = \begin{cases} 1 & \text{jos tiheän tason piste } j \text{ on harvan tason piste } i, \\ \frac{1}{k} & \text{jos tiheän tason piste } j \text{ on harvan tason pisteen } i \text{ naapuri} \\ & \text{ja sillä on } k \text{ harvan tason naapuripistettä,} \\ 0 & \text{muuten} \end{cases} \quad (3.22)$$

3.6 Menetelmän analysointia

Monihilamenetelmä on yksi tehokkaimmista ratkaisijoista elliptisille yhtälöille (Gholami ym. 2015). Monihila-algoritmillä iteraatioiden määrä on mahdollista pitää vakiona, vaikka ratkaistavan ongelman koko kasvaisi. Tämän vuoksi menetelmä on erittäin tarpeellinen ja hyödyllinen suurilla tehtävillä. Tehtävän laskentaa voidaan viedä harvemmille tasoille, joilla riittävän tarkan ratkaisun saavuttamiseen tarvitaan vähemmän iteraatiokierroksia ja laskeminen on nopeampaa.

Menetelmän eri vaiheita tarkastelemalla voidaan selvittää, kuinka paljon laskennallista työtä tarvitaan. Laskennallinen työ tarkoittaa ratkaisun laskemiseen tarvittavan

työn määrää, jota voidaan mitata esimerkiksi aritmeettisten operaatioiden määrällä. Monihilamenetelmän laskennallinen työ W_l , tasolla l , voidaan määrittellä rekursiivisesti (Trottenberg, Oosterlee ja Schüller 2001)

$$W_1 = W_1^0 + W_0, \quad W_{m+1} = W_{m+1}^m + \gamma m W_m, \quad m = 1, \dots, l-1, \quad (3.23)$$

jossa W_{m+1}^m tarkoittaa työtä, jota tarvitaan kahden tason, m ja $m+1$ välillä, W_0 tarkoittaa työtä, jota tarvitaan tarkkaan ratkaisemiseen harvimmalla tasolla 0 ja γ tarkoittaa iteraatioiden määrää tasolla. Jos γ on riippumaton m :sta, voidaan määrittellä

$$W_l = \sum_{m=1}^l \gamma^{l-m} W_m^{m-1} + \gamma^{l-1} W_0, \quad l \geq 1. \quad (3.24)$$

Arvioista W_l nähdään, että aritmeettisten operaatioiden määrä yhdellä monihilasyklillä on verrannollinen hilapisteiden määrään tiheimmällä tasolla kun $\gamma \leq 3$. Tämä tarkoittaa, että monihilamenetelmät saavuttavat riittävän tarkan ratkaisun $O(N)$ operaatiolla, kun N on tuntemattomien muuttujien määrä.

Ratkaistaessa harvan tason ongelmaa käyttäen V-sykliä tarvitaan käytännössä vain yksi iteraatio jokaisella tasolla, ja ongelma ratkaistaan tarkasti vain harvimmalla tasolla. Tällöin ongelma on niin pieni, että sen ratkaisemiseen vaadittava työ on mitätön. Erityisesti tiheillä tasoilla kannattaakin tehdä vain yksi iteraatiokierros. Ainut vaadittava työ on itse asiassa tasoittaminen eri tasoilla ja koska näiden lukumäärä on suoraan verrannollinen laskentatasojen määrään, on aritmeettisten operaatioiden määrä $O(N)$ (Yair 2008).

4 Syklinen reduktio

Syklinen reduktio on numeerinen menetelmä yhtälöryhmien ratkaisemiseen. Sen perusideana on joka askeleella eliminoida parittomat rivit yhtälöstä

Yhtälö (2.8) matriisimuodossa

$$\begin{bmatrix} D & I & & \\ I & D & I & \\ & \ddots & \ddots & \ddots \\ & & I & D \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \quad (4.1)$$

jossa D on $n \times n$ tridiagonaalinen matriisi, I on $n \times n$ identtinen matriisi ja $N = n \times n$. Kirjoitetaan auki matriisitulo, jolloin saadaan yhtälöt

$$u_{i-2} + Du_{i-1} + u_i = f_{i-1}, \quad (4.2)$$

$$u_{i-1} + Du_i + u_{i+1} = f_i, \quad (4.3)$$

$$u_i + Du_{i+1} + u_{i+2} = f_{i+1}, \quad (4.4)$$

jossa $i = 2, 4, \dots, N-1$ ja $u_0 = u_{N+1} = 0$. Kerrotaan seuraavaksi yhtälö (4.3) matriisilla $-D$ ja lasketaan rivit yhteen. Näin saadaan uusi yhtälöryhmä kokoa $2^{k-1} - 1$, josta on eliminoitu pois kaikki parittomat rivit (Sweet 1977).

$$u_{i-4} + D^{(1)}u_{i-2} + u_i = f_{i-2}^{(1)}, \quad (4.5)$$

$$u_{i-2} + D^{(1)}u_i + u_{i+2} = f_i^{(1)}, \quad (4.6)$$

$$u_i + D^{(1)}u_{i+2} + u_{i+4} = f_{i+2}^{(1)}, \quad (4.7)$$

jossa $i = 4, 8, \dots, N-4$ ja koska $T = I$

$$D^{(1)} = 2I - D^2, \quad (4.8)$$

$$f_i^{(1)} = f_{i-1} - Df_i + f_{i+1}. \quad (4.9)$$

Kun jatketaan parittomien rivien eliminointia samoin, saadaan kaavat (Rossi, Heikkola ja Toivanen 1998)

$$D^{(r)} = 2I - (D^{(r-1)})^2, \quad r = 1, \dots, k, \quad (4.10)$$

$$f_i^{(r)} = f_{i-2h}^{(r-1)} - f_{i-h}^{(r-2)} + f_i^{(r-1)} - f_{j+h}^{(r-2)} + f_{j+2h}^{(r-1)} + (D^{r-1})^{-1}(f_{i-3h}^{(r-2)} - f_{i-2h}^{(r-1)} + f_{i-h}^{(r-2)} - 2f_i^{(r-1)} + f_{i+h}^{(r-2)} - f_{i+2h}^{(r-1)} + f_{i+3h}^{(r-2)}), \quad (4.11)$$

jossa $h = 2^{r-2}$. Lisäksi $f_i^{(0)} = f_i$ ja $D^{(0)} = D$. Saadulla kaavalla voidaan eliminoida pois parittomia rivejä, kunnes jäljellä on enää yksi rivi. Tämän jälkeen aloitetaan takaisinsijoitusvaihe, jossa vaiheita edetään toiseen suuntaan, ratkaisten jokaisella askeleella parittomat rivit. Ratkaisemalle u_i kaavasta (4.3) saadaan

$$u_i = D^{-1}(f_i - u_{i-1} - u_{i+1}), \quad (4.12)$$

josta voidaan johtaa kaava parillisten rivien ratkaisemiseksi

$$u_i = \frac{1}{2}(f_{i-2h}^{(r-1)} + f_{i+2h}^{(r-1)} - f_i^{(r)}) + (D^{(r)})^{-1}(f_i^{(r)} - u_{i-4h} - u_{j+4h}). \quad (4.13)$$

Muodostetut matriisit $D^{(r)}$ ovat polynomeja astetta 2^r alkuperäisestä matriisista D (Rossi, Heikkola ja Toivanen 1998). Merkitään nyt polynomien juuria $\lambda_i^{(r)}$, jolloin jakamalla $D^{(r)}$ tekijöihin, voidaan kirjoittaa yhtälö (4.10) muodossa

$$D^{(r)} = -\prod_{i=1}^{2^r} (D - \lambda_i^{(r)} I) \quad (4.14)$$

Nyt ratkaisemalla x_{2^r} yhtälöstä

$$(D - \lambda_i^{(r)} I)x_i = x_{i-1}, \quad i = 1, 2, \dots, 2^r \quad (4.15)$$

saadaan etsitty ratkaisu $x_{2^r} = u$.

Tätä menetelmää kutsutaan pariton-parillinen-menetelmäksi. Kyseisessä menetelmässä ei tarvitse laskea ja tallentaa matriiseja $D^{(r)}$, vaan ainoastaan polynomien juuret tarvitaan. Jos matriisin A rivien määrä on $N = 2^p$, tarvitaan p reduktio-askelta, jotta saadaan yhtälö, jossa on vain yksi tuntematon u_N . Tämän ratkaisemiseksi pitää ratkaista N yhtälöä. Jokaisella reduktio-askeleella syntyy $N/2$ yhtälöä. Vastaavasti takaisinsijoitus-askeleita on p ja jokaisella askeleella ratkaistaan $N/2$ yhtälöä. Tällöin ratkaistavien yhtälöiden määrä on yhteensä $N(p+1) = N \log_2 N + N$. Matriisin D^{-1} jokaisella rivillä on ei-nollia alkioita huomattavasti rivien lukumäärää vähemmän. Jos matriisin D rivien määrä on $M = 2^r$, niin aritmeettisten operaatioiden määrä jokaisen u_i ratkaisemiseen kaavasta (4.12) on kertaluokkaa M . Tällöin menetelmän aritmeettisten operaatioiden määrä on kertaluokkaa $MN \log_2 N$ (Swarztrauber ja Sweet 1996).

5 Menetelmien vertailua

Luvussa 3 todettiin, että monihilamenetelmän aikavaativuus on luokkaa $O(N)$, kun N on tuntemattomien muuttujien määrä. Monihila-menetelmän osat, joita ovat residuaalin laskeminen, rajoittaminen ja prolongaatio, ovat rinnakkaistettavissa (McBryan ym. 1990). Myös tasoittamiseen käytetyt perusmenetelmät, kuten Jacobi, ovat luonnostaan rinnakkaistuvia (Göddeke ja Strzodka 2011). Monihilamenetelmä ei vaadi tietynlaista verkkoa, vaan sitä voidaan soveltaa esimerkiksi kolmioelementteihin (Mavriplis, Martinelli ja Jameson 1989) tai kokonaan rakenteettomille, epä-säännöllisille hiloille. Se ei myöskään vaadi mitään tiettyä diskreetointimenerelmää ja sitä voidaan soveltaa myös suoraan epälineaarisisille ongelmille (Trottenberg, Oosterlee ja Schüller 2001).

Luvussa 4 esitellyn syklinen reduktio -menetelmän aikavaativuus on luokkaa $O(N^2 \log_2 N)$. Laskettaessa matriiseja $D^{(r)}$ kaavasta (4.10) muisti täyttyy nopeasti, vaikka D on tridiagonaalinen. Käyttämällä parillinen-pariton-menetelmää, säästetään muistin tilaa, koska vain polynomien juuret täytyy tallentaa muistiin matriisien sijaan. Tällöin menetelmän muistiin tallentamiseen tarvitaan vektori, jonka pituus on N .

Koska jokaisella rekursioaskeleella tapahtuvat parittomien rivien eliminoinnit ovat toisistaan riippumattomia, syklinen reduktio soveltuu helposti rinnakkaistettavaksi. Tässä menetelmässä esitellyt versiot ovat rinnakkaistettavuuden kannalta kohtalaisen hyviä. Rivien määrä kuitenkin puolittuu jokaisella askeleella, mutta tehtävä työ pysyy kokoajan lähes samana. Tämä tuottaa ongelmia, koska kunkin rivin laskemiseksi ratkaistavat yhtälöryhmät ratkaistaan peräkkäin ja siten rinnakkaisten tehtävien määrä puolittuu rivien määrän puolittuessa (Myllykoski 2010). Alun perin syklinen reduktio -menetelmä on rajoitettu suorakulmaiselle alueelle, mutta on sittemmin laajennettu huomattavasti laajemmille ongelmille (Swarztrauber 1977), kuten tietyille epä-säännöllisille alueille (Buzbee ym. 1971). Sitä voidaan myös hyödyntää epälineaaristen ongelmien ratkaisemisessa (Bini ja Golub 2009).

6 Yhteenveto

Tutkielmassa esiteltiin Poissonin yhtälö sekä sen diskretointi. Diskretoitu muoto voidaan ratkaista erilaisilla numeerisilla menetelmillä, joista käytiin läpi tässä tutkielmassa kaksi. Monihilamenetelmä on iteratiivinen, nopea ja hyvin rinnakkais-tettavaksi soveltuva. Monihilamenetelmä perustuu tiheämpien ja harvempien taso- jen hyödyntämiseen laskennassa ja on yksi tehokkaimmista ratkaisijoista elliptisil- le yhtälöille. Menetelmän asymptoottinen aikavaativuus on $O(N)$. Monihilamene- telmät voidaan jakaa geometrisiin ja algebrallisiin, joista tässä tutkielmassa on esi- teltty algebrallinen monihilamenetelmä sekä sen muunnos, graafipohjainen moni- hilamenetelmä. Algebralliselle monihilaratkaisijalle ei tarvitse viedä tietoa verkon geometriasta vaan se tarvitsee vain alkuperäisen yhtälöryhmän. Syklinen reduktio -menetelmä on suora, nopea ja myös melko hyvin rinnakkaistettavaksi soveltuva. Menetelmän asymptoottinen aikavaativuus on $O(N^2 \log_2 N)$. Ensimmäisenä esitelty versio käyttää paljon tietokoneen muistia, mutta toisessa versiossa hyödynnettiin polynomien juuria ja muistia tarvitsee käyttää vain N pituisen vektorin verran.

Nopean Fourier-muunnos -menetelmä on sekä syklistä reduktiota, että monihila- menetelmää hitaampi. Myös Fourier-muunnoksiin perustuvat menetelmät ovat hel- posti rinnakkaistettavia. Monihilamenetelmää kannattaa käyttää ratkaistaessa suu- ria tehtäviä, sillä laskentaa voidaan helposti viedä harvemmille tasoille ja sen iteraa- tioiden määrä voidaan pitää vakiona riippumatta yhtälöryhmän koosta. Sykliseen reduktioon perustuvaa pariton-parillinen -menetelmää voidaan hyödyntää esimer- kiksi jos halutaan säästää tietokoneen muistia.

Lähteet

Bini, D., ja G. Golub. 2009. "The cyclic reduction algorithm: from Poisson equation to stochastic processes and beyond". *Numerical algorithms* 51 (1): 23–60.

Bolten, M. 2010. *Multigrid methods for long-range interactions*, toimittanut Godehard Sutmann, Paula Gibbon ja Thomas Lippert, 116–130. Forschungszentrum Jülich GmbH Zentralbibliothek.

Buzbee, B., F. Dorr, J. George ja G. Golub. 1971. "The direct solution of the discrete Poisson equation on irregular regions". *SIAM journal on numerical analysis* 8 (4): 722–736.

Esfahanian, V., B. Baghapour, M. Torabzadeh ja H. Chizari. 2008. "An efficient GPU implementation of cyclic reduction solver for high-order compressible viscous flow simulations". *Computers and fluids* 92:160–171.

Gholami, A., D. Malhotra, H. Sundar ja G. Biros. 2015. "FFT, FMM, or multigrid? A comparative study of state-of-art Poisson solvers in the unit cube". *ArXiv e-prints*.

Göddecke, D., ja R. Strzodka. 2011. "Cyclic reduction tridiagonal solvers on GPUs applied to mixed-precision multigrid". *IEEE transactions on parallel and distributed systems* 22 (1): 22–32.

Jeon, I., K. Choi, T. Kim, B. Choi ja H. Ko. 2013. "Constrainable multigrid for cloth". *Computer graphics forum* 32 (7): 31–39.

Kamath, G., L. Shi, E. Chow ja W. Song. 2015. "Distributed Multigrid Technique for Seismic Tomography in Sensor Networks". Teoksessa *Big Data Computing and Communications*, toimittanut Yu Wang, Hui Xiong, Shlomo Argamon, XiangYang Li ja JianZhong Li, 9196:297–310. Lecture Notes in Computer Science. Springer International Publishing.

Mavriplis, D., L. Martinelli ja A. Jameson. 1989. "Multigrid solution of the Navier-Stokes equations on triangular meshes". *AIAA Journal* 28 (8): 1415–1425.

- McBryan, O., P. Frederickson, J. Linden, A. Shuller, K.I Solchenbach, K. Stuben, C. Thole ja U. Trottenberg. 1990. "Multigrid methods on parallel computers - a survey on recent developments". *Computer Science Technical Reports*, numero 485.
- Myllykoski, M. 2010. "Rinnakkaistuvat nopeat Poisson-ratkaisijat". Kandidaatintutkielma, Jyväskylän yliopisto.
- . 2011. "Separoituvien yhtälöryhmien ratkaiseminen ohjelmoitavalla näyttönohjaimella". Pro gradu -tutkielma, Jyväskylän yliopisto.
- Pennanen, A. 2010. "A graph-based multigrid with applications". Väitöskirja, Jyväskylän yliopisto.
- Quarteroni, A., R. Sacco ja F. Saleri. 2007. *Numerical mathematics*. Springer.
- Rossi, T., E. Heikkola ja J. Toivanen. 1998. "Fast direct solvers and the fictitious domain method with separable preconditioners". Lecture notes, Jyväskylän yliopisto.
- Saraniti, M., A. Rein, G. Zandler, P. Vogl ja P. Lugli. 1996. "An efficient multigrid Poisson solver for device simulations". *Computer-aided design of integrated circuits and systems, IEEE transactions on* 15 (2): 141–150.
- Siikonen, T. 2014. "Laskennallisen virtausmekaniikan ja lämmönsiirron jatkokurssi". Luentomoniste, Aalto-yliopisto.
- Stümer, M., H. Köstler ja U. Rüde. 2007. "A fast full multigrid solver for applications in image processing". *Numerical linear algebra with applications* 15 (2-3): 187–200.
- Swarztrauber, P. 1977. "The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle". *SIAM Review* 19 (3): 490–501.
- Swarztrauber, P., ja R. Sweet. 1996. *The Fourier and Cyclic Reduction Methods for Solving Poisson's Equation*. John Wiley / Sons, Inc.
- Sweet, R. 1977. "A cyclic reduction algorithm for solving block tridigonal systems of arbitrary dimension". *SIAM Journal on Numerical Analysis* 14 (4): 706–720.
- Trottenberg, U., C. Oosterlee ja A. Schüller. 2001. *Multigrid*. Academic Press.

Wei, L., X. Yanfu, H. Zhongjie, H. Guijun ja L. Kexiu. 2008. "Application of the multigrid data assimilation scheme to the China Seas' temperature forecast". *Journal of atmospheric and oceanic technology* 25 (2): 2106–2116.

Yair, S. 2008. *Matrix-based multigrid : theory and applications*. Springer Science+Business Media.

Yousef, S. 2000. *Iterative methods for sparse linear systems*. Society for Industrial / Applied Mathematics.