

Funktionaalisia a posteriori virhearvioita
Uzawan tyyppisille menetelmille
kokoonpuristumattomien virtausten
tapauksessa

Marjaana Nokka

Matematiikan pro gradu

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
Kevät 2012

Tekijä	Marjaana Nokka
Sähköposti	marjaana.nokka@jyu.fi
Otsikko	Funktionaalisia a posteriori virhearvioita Uzawan tyyppisille menetelmille kokoonpuristumattomien virtausten tapauksessa
English title	A posteriori error estimates for Uzawa type methods in the theory of incompressible fluids
Työ	Matematiikan pro gradu -tutkielma
Ohjaajat	Immanuel Anjam, Sergey Repin, Xiao Zhong
Sivumäärä	73

Tiivistelmä: Stokesin yhtälöllä voidaan kuvata nesteiden ja kaasujen liikettä, jos liike on yksiulotteista tai hidasta. Stokesin yhtälö on yksinkertaistettu ja linearisoitu versio Navier-Stokesin yhtälöistä. Tässä tutkielmassa keskitytään kokoonpuristumattomiin ja viskoottisiin nesteisiin ja kaasuihin. Kokoonpuristumattomuus tarkoittaa sitä, että nesteen tai kaasun tiheys ei muutu ajan suhteen. Viskoottisuus taas tarkoittaa sitä, että nesteillä ja kaasuilla on sisäistä kitkaa, joka muodostuu, kun aineen osat liikkuvat toistensa suhteen.

Vaikka osittaisdifferentiaaliyhtälöiden tutkimus on edennyt viime vuosisadalla hyvinkin nopeasti, on analyttisen ratkaisun löytäminen vaikeaa tai lähes poikkeuksetta mahdotonta. Tapauksissa, joissa analyttinen ratkaisu löydetään kohtuullisella vaivalla, on yleensä tehty epärealistisen suuria yksinkertaistuksia tai yhtälö on alkujaan keinotekoinen. Jotta epärealistisilta yksinkertaistuksilta voitaisiin välttyä, on osittaisdifferentiaaliyhtälöille kehitetty monia numeerisia ratkaisumenetelmiä, kuten esimerkiksi tässä tutkielmassa hyödynnetty elementtimenetelmä. Numeeristen menetelmien käyttö tuottaa osittaisdifferentiaaliyhtälölle likimääräisen ratkaisun, joka hyvin harvoin vastaa tarkkaa ratkaisua. Likimääräiseen ratkaisuun sisältyy siis aina jonkunlainen virhe.

Funktionaalisilla a posteriori virhearvioilla voidaan arvioida luotettavasti sitä virhettä, joka tehdään ratkaistaessa osittaisdifferentiaaliyhtälöitä numeerisesti. Tässä tutkielmassa malliongelmana käytetään yleistettyä Stokesin yhtälöä. Tälle ongelmalle on jo aiemmin johdettu a posteriori virhearviot, tässä tutkielmassa johdetaan vastavat virhearviot ottaen huomioon Uzawa-algoritmin erityispiirteet. Uzawa-algoritmi on suosittu menetelmä likiarvotarkaisujen löytämiseen satulapisteongelmille. Lisäksi tarkastellaan Uzawa-algoritmin konvergenssia malliongelman tapauksessa.

Seuraavaksi tutkimme approksimaatiovirhettä, joka muodostuu, kun ääretönulotteinen ongelma korvataan äärellisulotteisella ongelmalla. Tämän myötä virheen ylärajaan (majorantiin) tulee vapaita parametreja, jotka tulee minimoida mahdollisimman tarkan ylärajan löytämiseksi. Numeerisessa osuudessa ratkaistaan malliongelma elementtimenetelmällä MINI-elementtiä hyödyntäen ja minimoidaan majorantin vapaat parametrit elementtimenetelmällä Raviart-Thomas-elementtejä hyödyntäen. Lopuksi vahvistetaan johdettujen majoranttien toimivuus numeerisilla testeillä. Numeerisia testejä varten on koottu molemmille elementtityypeille osittaisdifferentiaaliyhtälöratkaisijat.

Avainsanat: funktionaalinen, a posteriori, virhearvio, yläraja, majorantti, ODY, Stokes, FEM, Raviart-Thomas, MINI, Uzawa

Sanasto

a posteriori	jälkeen
a priori	ennen
FEM	elementtimenetelmä (engl. <i>finite element method</i>)
majorantti	ks. yläraja
tarkka	elementtiverkkoa tihentämällä tarkka yläraja antaa mielivaltaisen tarkan arvion virheestä
yläraja	suure, joka on aina suurempi tai yhtä suuri kuin tarkka virhe

Matemaattiset merkinnät

\mathbb{R}^n	n -ulotteinen euklidinen reaaliavaruus
$x \cdot y$	vektorien x ja y sisätulo
$\tau : \sigma$	tensorien τ ja σ sisätulo
δ_{ij}	Kroneckerin delta
∇	gradienttioperaattori
$\mathbb{1}$	yksikkötensori
div	divergenssioperaattori vektoriarvoisille funktioille
Div	divergenssioperaattori tensoriarvoisille funktioille
$\ \cdot \ $	L_2 -normi
$\ \cdot \ $	energianormi
Ω	alue \mathbb{R}^n :ssä
F_K	affiinikuvaus referenssielementiltä \hat{K} verkon elementille K
B_K	funktion F_K matriisiosa
b_K	funktion F_K vektoriosa

Esipuhe

Tämän tutkielman teko alkoi vuosi sitten, kun sain mahdollisuuden työskennellä Prof. Sergey Repinin tutkimusryhmässä. Prosessin aikana olen saanut perusteellisen perehdytyksen virheen estimointiin ja elementtimenetelmän käyttöön. Hyvien tutkimustulosten myötä kirjoitimme pro gradu -tutkielman aiheesta artikkelin [1].

Haluaisin kiittää Prof. Sergey Repiniä mahdollisuudesta tehdä pro gradu -tutkielmani tästä aiheesta. Lisäksi haluan kiittää ohjaajiani Prof. Xiao Zhongia matemaattisen puolen tarkastamisesta, sekä Immanuel Anjamia perusteellisesta perehtymisestä tutkielmani aiheeseen ja erityisesti avusta ratkaisijoiden toteuttamisessa. Kiitos myös Olli Malille vastauksista moniin kysymyksiin. Lopuksi haluaisin kiittää Pirkko ja Veikko Mäkelän säätiötä rahallisesta tuesta.

Sisältö

1	Johdanto	1
2	Merkintöjä, määritelmiä ja perustuloksia	3
2.1	Differentiaalioperaattoreita	3
2.2	Funktioavaruuksista	4
2.3	Epäyhtälöitä	6
3	Yleistetty Stokesin yhtälö	8
3.1	Heikko muoto ja variaationaalimuoto	8
3.2	Satulapisteongelma	9
3.3	Kohtisuoruuslause	9
4	Uzawa-algoritmi	12
4.1	Algoritmi	12
4.2	Uzawa-algoritmin konvergenssi	13
5	Funktionaaliset a posteriori virhearviot	17
5.1	Nopeuskentän virhemajorantti	17
5.2	Uzawa-algoritmin askelkohtainen virhe	23
5.3	Majorantin globaali minimointi	29
6	Numeriikkaa	30
6.1	Vakioiden määrittäminen	30
6.2	Elementtimenetelmä	31
6.3	Osittaisdifferentiaaliyhtälön ratkaisijan muodostaminen	32
6.4	Affinikuvauksen muodostaminen	33
6.5	Kantafunktioiden muodostaminen	34
6.5.1	MINI-elementti	34
6.5.2	Raviart-Thomas-elementti	36
6.6	Numeeriset tulokset	38
6.6.1	Tarkka ratkaisu tunnetaan	38
6.6.2	Vapaiden parametrien minimointi	43
6.6.3	Tarkkaa ratkaisua ei tunneta	44
7	Johtopäätökset	46
	Kirjallisuus	47

A	Lähdekoodit	50
A.1	Malliongelman ratkaiseminen (solverMINI.m)	50
A.2	Tensoriarvoinen minimointiongelma (solverRT0t.m)	57
A.3	Majorantin laskeminen	60

1 Johdanto

Suurin osa maapallosta on veden peitossa ja koko maapallo on päällystetty ilmalla, joten virtausmekaniikalla on tärkeä osa monien fysikaalisten ilmiöiden mallintamisessa. Virtausmekaniikka tutkii fluidien, eli nesteiden ja kaasujen, käyttäytymistä erilaisissa olosuhteissa. Erilaisten virtausten mallintaminen on tärkeää esimerkiksi lääketieteessä, meteorologiassa, aerodynamiikassa ja putkistosuunnittelussa. Näitä virtauksia voidaan mallintaa ns. Navier-Stokesin yhtälöillä, jotka osittaisdifferentiaaliyhtälöistä koostuen kuvaavat nesteen tai kaasun liikettä. Yksinkertaisesta muotoilustaan ja monista sovellusalueistaan huolimatta Navier-Stokesin yhtälöt ovat erittäin vaikea yhtälöryhmä ratkaistavaksi.

Stokesin yhtälö on tärkeä yksinkertaistettu versio Navier-Stokesin yhtälöistä. Stokesin yhtälöä voidaan käyttää muun muassa silloin, kun virtaus on yksiulotteista tai ns. ryömivää eli hyvin hidasta [26]. Tällaista virtausta kutsutaan Stokesin virtaukseksi. Stokesin virtaus on nimetty George Stokesin ¹ mukaan. Stokesin virtaukselle on tyypillistä, että sen advektiiviset inertiaivoimat ovat pieniä verrattuna viskoottisiin voimiin. Lisäksi fluidin Reynoldsin luku on reilusti alle yhden. Reynoldsin luku määritellään fluidin nopeuden U , leveyden L , tiheyden ρ ja viskositeetin ν avulla seuraavasti:

$$\text{Re} = \frac{\rho UL}{\nu}.$$

Jotta Reynoldsin luku olisi hyvin pieni, tulee U :n tai L :n olla verrattaen pieniä, tai viskositeetin ν verraten suuri. Materiaalikohtaisena parametrina viskositeetti kuvaa fluidin kykyä vastustaa virtausta. Viskositeetin voidaan ajatella kuvaavan nesteen ”paksuutta”, esimerkiksi vesi on ”ohuempaa” kuin kasviöljy. Myös veri virtaa joissakin kehon osissa Stokesin virtauksen tyyppisesti.

Tässä työssä virtauksen oletetaan olevan Stokesin virtauksen tyyppistä. Lisäksi oletetaan massan säilymlaki päteväksi. Massan säilymlain mukaan suljetun systeemin massa pysyy vakiona ajan suhteen. Massan säilymlaki on yksi virtausmekaniikan perusoletuksista. Sen lisäksi oletetaan yleensä myös energian ja liikemäärän säilyminen.

Vaikka osittaisdifferentiaaliyhtälöiden teorian kehitys on ollut nopeaa viime vuosisadalla, on analyttisten ratkaisujen etsiminen vaikeaa ja lähes poikkeuksetta mahdotonta. Siksi osittaisdifferentiaaliyhtälöille on kehitetty usei-

¹George Gabriel Stokes (1819-1903)

ta numeerisia ratkaisumenetelmiä, joilla päästään hyvin lähelle oikeaa ratkaisua. Tässä työssä käsiteltävä numeerinen menetelmä on elementtimenetelmä (eng. *finite element method*, lyh. *FEM*), joka on hyvin tehokas menetelmä osittaisdifferentiaaliyhtälöiden ratkaisemisessa. Stokesin yhtälö muotoillaan satulapisteongelmaksi ja sen likiarvoratkaisuja etsitään iteratiivisesti Uzawa-algoritmillä, joka on tehokas menetelmä satulapisteongelmille.

Tässä työssä tutkitaan a posteriori virhearvioita yleistetyille Stokesin yhtälölle. Tutkielman toisessa luvussa esitetään tarvittavia matemaattisia merkintöjä, määritelmiä ja muutamia olennaisia perustuloksia, joiden todistukset sivutetaan kirjallisuusviittauksin. Kolmannessa luvussa esitellään tutkielman malliongelma Stokesin yhtälö. Uzawa-algoritmiin ja sen konvergenssiin tutustutaan luvussa neljä. Viidennessä luvussa päästään tutkielman pääaiheeseen, eli a posteriori virhearvioiden johtamiseen. Kuudennessa luvussa käsitellään numeerisia tuloksia, esitellään osittaisdifferentiaaliyhtälöratkaisijan toiminta ja esitetään kirjallisuudesta löytyviä arvioita virhearvioissa esiintyville aluekohtaisille vakioille. Liitteisiin on koottu tärkeimmät ohjelmakoodit, joita numeeristen tulosten laskennassa on käytetty.

2 Merkintöjä, määritelmiä ja perustuloksia

Tässä luvussa esitellään myöhemmin tarvittavia matemaattisia merkintöjä ja perustuloksia.

Vektoriavaruudessa \mathbb{R}^n merkitään kanonista kantaa merkinnällä $e_1 = (1, 0, \dots, 0)$, $e_2 = (0, 1, 0, \dots, 0)$, \dots , $e_n = (0, \dots, 0, 1)$. Avaruuden \mathbb{R}^n vektoria merkitään $x = (x_1, \dots, x_n)$. Kahden vektorin sisätuloa merkitään symbolilla \cdot , esimerkiksi

$$x \cdot y := \sum_{i=1}^n x_i y_i.$$

Kanonisen kannan kantavektorien sisätuloksi saadaan

$$e_i \cdot e_j = \delta_{ij},$$

missä Kroneckerin delta $\delta_{ii} = 1$, ja $\delta_{ij} = 0$ jos $i \neq j$. Määritellään tensoriavaruus $\mathbb{M}^{n \times n}$ asettamalla

$$\mathbb{M}^{n \times n} := \mathbb{R}^n \times \mathbb{R}^n.$$

Avaruuden $\mathbb{M}^{n \times n}$ alkioille τ ja σ sisätulo määritellään

$$\tau : \sigma := \sum_{i,j=1}^n \tau_{ij} \sigma_{ij}.$$

2.1 Differentiaalioperaattoreita

Olkoon funktio $f : \Omega \rightarrow \mathbb{R}$. Tällöin funktion f gradientti on vektori

$$\nabla f := \left(\frac{\partial f}{\partial x_1}; \dots; \frac{\partial f}{\partial x_n} \right).$$

Olkoon funktio $u : \Omega \rightarrow \mathbb{R}^n$, $u(x) = (u_1(x), u_2(x), \dots, u_n(x))$. Tällöin funktion u gradientti on funktio

$$\nabla u := \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \dots & \frac{\partial u_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial u_1}{\partial x_n} & \dots & \frac{\partial u_n}{\partial x_n} \end{bmatrix}$$

ja divergenssi on funktio

$$\operatorname{div} u := \sum_{i=1}^n \frac{\partial u_i}{\partial x_i}.$$

Tensorille $\tau \in \mathbb{R}^n \times \mathbb{R}^n$ divergenssi määritellään riveittäin

$$\operatorname{Div} \tau := \left[\sum_{i=1}^n \frac{\partial \tau_{1i}}{\partial x_i}; \dots; \sum_{i=1}^n \frac{\partial \tau_{ni}}{\partial x_i} \right].$$

2.2 Funktioavaruuksista

Symbolilla Ω merkitään \mathbb{R}^n :n yhtenäistä ja rajoitettua osajoukkoa jolla on Lipschitz-jatkua reuna $\partial\Omega$. Olkoon $\omega \subset \Omega$ avoin joukko. Joukon ω sulkeumaa merkitään $\bar{\omega}$. Funktion $f : \Omega \rightarrow \mathbb{R}$ sanotaan kuuluvan luokkaan C^k , jos se on k -kertaa jatkuvasti derivoituva, eli jos sen jokainen derivaatta $D^\alpha u$, $\alpha \leq |k|$ on olemassa ja jatkuva. Tässä $\alpha = (\alpha_1, \dots, \alpha_n)$ on multi-indeksi siten, että $|\alpha| = \alpha_1 + \dots + \alpha_n$, ja

$$D^\alpha u := \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}.$$

Skalaariarvoisille funktioille käytetään merkintää $C^k(\Omega)$. Vektoriarvoisille funktioille w käytetään merkintää $C^k(\Omega, \mathbb{R}^n)$, jos w :n jokainen komponentti $w_i \in C^k(\Omega)$. Vastaavaa notaatiota tullaan käyttämään myös muiden avaruuksien kanssa.

Avaruus $L_2(\Omega)$ muodostuu funktioista, jotka ovat neliöintegroituvia Ω :ssa, siis

$$L_2(\Omega) := \{f : \Omega \rightarrow \mathbb{R} \mid \|f\|_{2,\Omega} < \infty\},$$

missä

$$\|f\|_{2,\Omega} := \left(\int_{\Omega} |f|^2 \, dx \right)^{1/2}.$$

Lisäksi määritellään

$$\tilde{L}_2(\Omega) := \{w \in L_2(\Omega) \mid \int_{\Omega} w \, dx = 0\}.$$

Avaruudelle L_∞ vastaavat määritelmät ovat

$$L_\infty(\Omega) := \{f : \Omega \rightarrow \mathbb{R} \mid \|f\|_{\infty,\Omega} < \infty\},$$

missä

$$\|f\|_{\infty, \Omega} := \operatorname{ess\,sup}_{x \in \Omega} |f(x)|.$$

Otetaan käyttöön merkinnät $\|w\| = \|w\|_{2, \Omega}$ ja $\|w\|_{\infty} = \|w\|_{\infty, \Omega}$. Vektori- ja tensoriarvoisille funktioille määritellään

$$\begin{aligned} L_2(\Omega, \mathbb{R}^n) &:= \{w : \Omega \rightarrow \mathbb{R}^n \mid w_i \in L_2(\Omega) \forall i = 1, \dots, n\}, \\ L_2(\Omega, \mathbb{M}^{n \times n}) &:= \{\tau : \Omega \rightarrow \mathbb{M}^{n \times n} \mid \tau_{ij} \in L_2(\Omega) \forall i, j = 1, \dots, n\}, \end{aligned}$$

ja näiden normi määritellään ja merkitään kuten reaaliarvoisen L_2 -funktion normi.

Funktioita, joilla on kaikkien kertalukujen derivaatat alueessa Ω , merkitään \mathcal{C}^{∞} . Kompaktikantajaiset funktiot $f \in \mathcal{C}^{\infty}$ muodostavat aliavaruuden \mathcal{C}_0^{∞} . Funktion f kantaja on sulkeuma niistä lähtöjoukon pisteistä, jossa funktion arvo poikkeaa nolasta, toisin sanoen

$$\operatorname{spt} f = \overline{\{x \in \Omega \mid f(x) \neq 0\}}.$$

Avaruudeksi $V(\Omega)$ määritellään Sobolev-avaruus $W_2^1(\Omega, \mathbb{R}^n)$. V on siis joukon

$$\{\phi \in \mathcal{C}^{\infty}(\Omega) : \|\phi\|_{1,2} < \infty\}$$

täydellistymä normin

$$\|\phi\|_{1,2} := \|\phi\| + \|\nabla \phi\|$$

suhteen. Tämä on yhtäpitävää sen kanssa, että $u \in L_2(\Omega, \mathbb{R}^n)$ ja on olemassa tensoriarvoinen funktio $\eta \in L_2(\Omega, \mathbb{M}^{n \times n})$ siten, että erälle funktiojonolle $(\phi_j) \in \mathcal{C}^{\infty}$ pätee $\|u - \phi_j\| \rightarrow 0$ ja $\|\eta - \nabla \phi_j\| \rightarrow 0$, kun $j \rightarrow \infty$. Tällöin funktiota η sanotaan u :n heikoksi gradientiksi. Vastaavasti määritellään heikko divergenssi. Jatkossa kaikki differentiaalioperaattorit oletetaan heikoiksi.

Merkinnällä V_0 tarkoitetaan joukon V sellaista osajoukkoa, jossa funktiot häviävät alueen Ω reunalla. Yhtäpitävästi $w \in L_2(\Omega, \mathbb{R}^n)$ siten, että erälle funktiojonolle $(\varphi_j) \in \mathcal{C}_0^{\infty}$ pätee $\|w - \varphi_j\|_{1,2} \rightarrow 0$ kun $j \rightarrow \infty$.

Divergenssivapaiden, avaruuteen V_0 :n kuuluvien funktioiden joukko muodostaa avaruuden $S_0(\Omega)$, siis

$$S_0(\Omega) := \{w \in V_0 \mid \operatorname{div} w = 0\}.$$

Määritellään avaruus

$$\Sigma(\Omega) := L(\Omega, \mathbb{M}^{n \times n})$$

ja tämän aliavaruutena edelleen

$$\Sigma(\text{Div}, \Omega) := \{\tau \in \Sigma \mid \text{Div} \tau \in L_2(\Omega, \mathbb{R}^n)\}.$$

2.3 Epäyhtälöitä

Myöhemmin tarvitaan seuraavia tunnettuja epäyhtälöitä. Hölderin ja Friedrichin epäyhtälöt on esitetty niissä avaruuksissa, joissa niitä tullaan käyttämään.

Lemma 2.3.1 (Hölderin epäyhtälö [11]). *Olkoon $f, g \in L_2(\Omega)$. Tällöin*

$$\int_{\Omega} |fg| \, dx \leq \|f\| \|g\|.$$

Lemma 2.3.2 (Youngin epäyhtälö [11]). *Olkoon a ja b ei-negatiivisia, ja p ja q positiivisia reaalilukuja siten, että $\frac{1}{p} + \frac{1}{q} = 1$. Tällöin*

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}.$$

Lemma 2.3.3 (Cauchy-Schwarzin epäyhtälö [12]). *Olkoon x ja y vektoreita avaruudessa \mathbb{R}^n . Tällöin*

$$\left| \sum_{i=1}^n x_i y_i \right|^2 \leq \sum_{j=1}^n |x_j|^2 \sum_{k=1}^n |y_k|^2.$$

Lemma 2.3.4 (Friedrichin epäyhtälö [18]). *Olkoon $\Omega \subset \mathbb{R}^n$ rajoitettu alue ja $u \in V_0(\Omega)$ funktio. Tällöin*

$$\|u\| \leq C_{\Omega} \|\nabla u\|,$$

missä vakio C_{Ω} riippuu vain alueesta Ω .

Olkoon $I = I_1 \times I_2 \times \dots \times I_n \subset \mathbb{R}^n$ siten, että $\Omega \subset I$. Friedrichin epäyhtälössä esiintyvällä vakiolla C_{Ω} voidaan määrittää arvo

$$C_{\Omega} = \sqrt{\pi^{-2}(a_1^{-2} + a_2^{-2} + \dots + a_n^{-2})^{-1}},$$

missä a_i on välin I_i pituus kaikilla $i = 1, 2, \dots, n$, ks [18].

Lemma 2.3.5. *Olkoon Ω rajoitettu alue siten, että sen reuna on Lipschitz-jatkua. Tällöin on olemassa positiivinen vakio κ_Ω (joka riippuu vain alueesta Ω) siten, että kaikille funktiolle $f \in \tilde{L}_2(\Omega)$ löytyy funktio $w \in V_0(\Omega)$, joka toteuttaa yhtälöt*

$$\operatorname{div} w = f$$

ja

$$\|\nabla w\| \leq \kappa_\Omega \|f\|. \quad (2.1)$$

Todistus. Ks [14, 13]. □

Joillekin yksinkertaisille alueille vakio κ_Ω on tunnettu, ks [4, 6, 10]. Tämä vakio on ns. Ladyzhenskaya-Babuska-Brezzi -ehdossa esiintvän vakion (LBB-vakio) käänteisluku.

Huomautus 2.3.6. *Lemmalla 2.3.5 on seuraava tärkeä sovellus: Olkoon $v \in V_0(\Omega)$ ja $\operatorname{div} v = g$. Tällöin on olemassa funktio $v_g \in V_0(\Omega)$ siten, että $\operatorname{div}(v - v_g) = 0$ ja*

$$\|\nabla v_g\| \leq \kappa_\Omega \|g\| = \kappa_\Omega \|\operatorname{div} v\|.$$

Tämä tarkoittaa, että on olemassa divergenssivapaa kenttä $v_0 = (v - v_g) \in S_0(\Omega)$ siten, että

$$\|\nabla(v - v_0)\| \leq \kappa_\Omega \|\operatorname{div} v\|.$$

3 Yleistetty Stokesin yhtälö

Olkoon Ω rajoitettu alue avaruudessa \mathbb{R}^n , $\partial\Omega$ sen reuna ja $f \in L_2(\Omega, \mathbb{R}^n)$. Tarkoitus on löytää nopeutta kuvaava vektoriarvoinen funktio $u \in V_0$, ja skalaarivoinen painetta kuvaava funktio $p \in \tilde{L}_2(\Omega)$, siten, että seuraava yhtälöryhmä toteutuu.

$$\begin{aligned} -\text{Div}\sigma + \mu u + \nabla p &= f && \Omega\text{:ssa} \\ \sigma &= \nu \nabla u && \Omega\text{:ssa} \\ \text{div}u &= 0 && \Omega\text{:ssa} \\ u &= 0 && \partial\Omega\text{:lla.} \end{aligned} \tag{3.1}$$

Tämä yhtälöryhmä kuvaa kokoonpuristumattoman aineen hidasta liikettä ja sitä kutsutaan yleistetyksi Stokesin yhtälöksi. Funktio ν kuvaa fluidin viskositeettia, ja μ voi olla paikan funktio. Funktioille $\nu, \mu \in L_\infty(\Omega)$ pätee

$$\begin{aligned} 0 < \underline{\nu} \leq \nu(x) \leq \bar{\nu} & \quad \forall x \in \bar{\Omega}, \\ 0 \leq \underline{\mu} \leq \mu(x) \leq \bar{\mu} & \quad \forall x \in \bar{\Omega}. \end{aligned}$$

Yhtälö $\text{div}u = 0$ tulee massan säilymislaista, jonka mukaan suljetun systeemin massa pysyy vakiona.

3.1 Heikko muoto ja variaationaalimuoto

Ongelman (3.1) variaationaalimuoto on

$$\inf_{v \in S_0(\Omega)} J(v) : J(v) := \int_{\Omega} \left(\frac{1}{2} \nu |\nabla v|^2 + \frac{1}{2} \mu |v|^2 - f v \right) dx. \tag{3.2}$$

Ongelman heikko muoto saadaan yhtälöstä (3.2) laskemalla Gateaux-derivaatta

$$\left. \frac{d}{dt} J(u + tw) \right|_{t=0},$$

jolloin heikko muoto on

$$\int_{\Omega} (\nu \nabla u : \nabla w + \mu u \cdot w) dx = \int_{\Omega} f \cdot w dx \quad \forall w \in S_0(\Omega). \tag{3.3}$$

Määritellään energianormi seuraavasti:

$$\|w\| := \left(\int_{\Omega} (\nu \nabla w : \nabla w + \mu w \cdot w) dx \right)^{1/2}. \tag{3.4}$$

3.2 Satulapisteongelma

Divergenssiehto $\operatorname{div} u = 0$ on numeerisesti hyvin vaikea toteuttaa. Siksi tarkastellaan ongelmaa, jossa etsitään $v \in V_0$ ja $q \in \tilde{L}_2(\Omega)$ siten, että

$$\begin{aligned} -\operatorname{Div} \sigma + \mu v + \nabla q &= f && \Omega:\text{ssa} \\ \sigma &= \nu \nabla v && \Omega:\text{ssa} \\ v &= 0 && \partial\Omega:\text{lla,} \end{aligned} \quad (3.5)$$

jonka heikko muoto on

$$\int_{\Omega} (\nu \nabla v : \nabla w + \mu v \cdot w - q \operatorname{div} w) \, dx = \int_{\Omega} f \cdot w \, dx. \quad (3.6)$$

Yhtälö (3.5) ja sen heikko muoto, yhtälö (3.6), ovat ekvivalentteja satulapisteongelman

$$\inf_{v \in V_0(\Omega)} \sup_{q \in \tilde{L}_2(\Omega)} L(v, q) \quad (3.7)$$

kanssa, missä ns. Langrangen muoto L on

$$L(v, q) := \int_{\Omega} \left(\frac{1}{2} \nu |\nabla v|^2 + \frac{1}{2} \mu |v|^2 - f v - q \operatorname{div} v \right) \, dx. \quad (3.8)$$

Ekvivalenttius on todettu kirjassa [7]. Helposti nähdään, että

$$\sup_{q \in \tilde{L}_2(\Omega)} L(v, q) = \begin{cases} J(v), & \text{jos } \operatorname{div} v = 0 \\ \infty & \text{muulloin.} \end{cases} \quad (3.9)$$

Satulapisteongelman tarkka ratkaisu on (u, p) , eli

$$L(v, p) \leq L(u, p) \leq L(u, q).$$

3.3 Kohtisuoruuslause

Tässä osiossa esitellään allekirjoittaneen ohjaajien osoittama myöhemmin tarvittava keskeinen tulos. Vastaavaa tulosta ei löydetty kirjallisuudesta, joten todistus on esitetty alla.

Lause 3.3.1. *Olkoon $(u, p) \in S_0(\Omega) \times \tilde{L}_2(\Omega)$ Stokesin yhtälön (3.3) toteuttava funktiopari ja $(v, q) \in V_0(\Omega) \times \tilde{L}_2(\Omega)$ Stokesin yhtälön (3.6) toteuttava funktiopari. Tällöin*

$$\|u - \phi\| \leq \|v - \phi\| \quad \forall \phi \in S_0(\Omega). \quad (3.10)$$

Todistus. Lauseen oletuksen nojalla tiedämme seuraavat yhtälöt päteviksi

$$\int_{\Omega} (\nu \nabla u : \nabla w + \mu u \cdot w) \, dx = \int_{\Omega} f \cdot w \, dx \quad \forall w \in S_0(\Omega),$$

$$\int_{\Omega} (\nu \nabla v : \nabla \tilde{w} + \mu v \cdot \tilde{w}) \, dx = \int_{\Omega} (f \cdot \tilde{w} + p \operatorname{div} \tilde{w}) \, dx \quad \forall \tilde{w} \in V_0(\Omega).$$

Olkoon $\tilde{w} \in S_0(\Omega)$ mielivaltainen. Tällöin divergenssitermi häviää, ja voimme kirjoittaa

$$\int_{\Omega} (\nu \nabla u : \nabla w + \mu u \cdot w) \, dx = \int_{\Omega} (\nu \nabla v : \nabla w + \mu v \cdot w) \, dx \quad \forall w \in S_0(\Omega).$$

Olkoon $\phi \in S_0(\Omega)$. Vähentämällä $\int_{\Omega} (\nu \nabla \phi : \nabla w + \mu \phi \cdot w) \, dx$ puolittain edellisestä yhtälöstä saamme

$$\begin{aligned} \int_{\Omega} (\nu \nabla(u - \phi) : \nabla w + \mu(u - \phi) \cdot w) \, dx \\ = \int_{\Omega} (\nu \nabla(v - \phi) : \nabla w + \mu(v - \phi) \cdot w) \, dx. \end{aligned}$$

Valitaan $w = u - \phi \in S_0(\Omega)$, jolloin Cauchy-Schwarzin epäyhtälön (lemma 2.3.3) nojalla saadaan

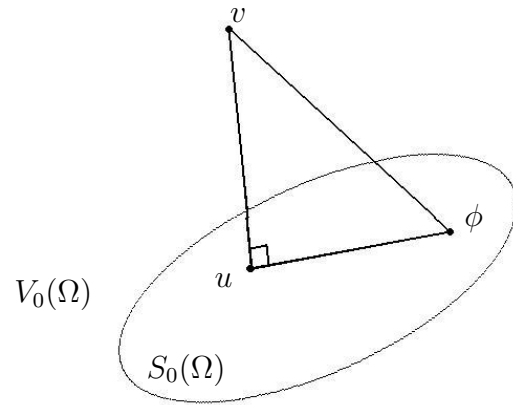
$$\begin{aligned} \|u - \phi\|^2 &= \int_{\Omega} \nu (\nabla(u - \phi))^2 + \mu(u - \phi)^2 \, dx \\ &= \int_{\Omega} (\nu^{1/2} \nabla(v - \phi) : \nu^{1/2} \nabla(u - \phi) + \mu^{1/2}(v - \phi) \cdot \mu^{1/2}(u - \phi)) \, dx \\ &\leq \|v - \phi\| \|u - \phi\|, \end{aligned}$$

josta saamme

$$\|u - \phi\| \leq \|v - \phi\|.$$

□

Huomautus 3.3.2. Yhtälön (3.10) muodosta nähdään, että $u - v \perp S_0(\Omega)$ (ks. kuva 1).



Kuva 1: Yleistetyn Stokesin yhtälön tarkan ratkaisun u , ja ratkaisun v suhtautuminen toisiinsa ja avaruuteen $S_0(\Omega)$.

4 Uzawa-algoritmi

Ratkaisun löytämisen helpottamiseksi turvaudutaan usein numeerisiin algoritmeihin, joilla päästään lähelle etsittyä ratkaisua. Käytännöllisiä Stokesin yhtälön ratkaisujen approksimoinnissa ovat esimerkiksi Uzawan algoritmiin pohjautuvat menetelmät, sekä Arrow-Hurwiczin algoritmi. Uzawan ja Arrow-Hurwiczin algoritmeista lisätietoa löytyy muun muassa Temamin kirjasta [25]. Tässä työssä käytetään approksimaatiomenetelmänä Uzawan algoritmia ja tutkitaan sen konvergenssia. Uzawan algoritmilla voidaan ratkaista Stokesin yhtälön kaltaisia satulapisteongelmia. Algoritmin perusperiaate on vuorotellen minimoida nopeuskenttää u ja maksimoida painekenttää p .

4.1 Algoritmi

Menetelmässä oleva parametri ρ kuvaa sitä, kuinka paljon sakotetaan massan säilymislain $\operatorname{div} u = 0$ rikkomisesta. Tavallisessa tilanteessa (kuten yhtälöä (3.5) käyttäessä) ei ρ :n valintaa voida kuitenkaan tehdä täysin mielivaltaisesti, vaan valintaa sitoo ehto $0 < \rho < 2 \min(\underline{\nu}, \underline{\mu})$.

Algoritmi 1 Uzawa-algoritmi

1. Tehdään paineelle alkuarvaus p^0 ja asetetaan $k = 0$.
2. Ratkaistaan u^k yhtälöstä

$$\begin{aligned} -\operatorname{Div} \sigma^k + \mu u^k + \nabla p^k &= f && \Omega\text{:ssa} \\ \sigma^k &= \nu \nabla u^k && \Omega\text{:ssa} \\ u^k &= 0 && \partial\Omega\text{:lla,} \end{aligned} \tag{4.1}$$

missä u^k tulkitaan ratkaisuksi heikossa mielessä.

3. Ratkaistaan

$$p^{k+1} = p^k - \rho \operatorname{div} u^k. \tag{4.2}$$

4. Jos lopetusehdot eivät täyty, asetetaan $k = k + 1$ ja siirrytään askeleeseen 2.
-

4.2 Uzawa-algoritmin konvergenssi

Lause 4.2.1. *Uzawan algoritmin (Algoritmi 1) likiarvoratkaisujen jono (u^k, p^k) konvergoi kohti Stokesin yhtälön tarkkaa ratkaisua, ts.*

$$\begin{aligned} u^k &\longrightarrow u & V:ssä \\ p^k &\longrightarrow p & \text{heikosti } L_2(\Omega):ssa, \end{aligned}$$

jos

$$0 < \rho < 2 \min(\underline{\nu}, \underline{\mu}). \quad (4.3)$$

Jos $\mu \equiv 0$, niin ehto on muotoa $0 < \rho < 2\underline{\nu}$.

Todistusta varten tehdään seuraavat määritelmät: Olkoon $u, v \in W_p^k(\Omega)$. Määritellään sisätulo

$$(u, v) := \int_{\Omega} u(x) \cdot v(x) \, dx.$$

Vastaavasti

$$(\nabla u, \nabla v) := \int_{\Omega} \nabla u(x) : \nabla v(x) \, dx.$$

Temam [25] osoittaa konvergenssin yleistetylle Stokesin yhtälölle tapauksessa $\nu \in \mathbb{R}_+$ ja $\mu \equiv 0$. Seuraavaksi esitettävä todistus mukailee tätä Temamin todistusta.

Lauseen 4.2.1 todistus. Sisätulomuotoa käyttäen saadaan ongelma heikkoon muotoon

$$(\nu \nabla u, \nabla v) + (\mu u, v) - (p, \operatorname{div} v) = (f, v) \quad \forall v \in V_0(\Omega). \quad (4.4)$$

Vastaavaan sisätulomuotoon saadaan myös Uzawan algoritmi:

$$(\nu \nabla u^k, \nabla v) + (\mu u^k, v) - (p^k, \operatorname{div} v) = (f, v) \quad \forall v \in V_0(\Omega). \quad (4.5)$$

Valitaan $v = u^k - u$ yhtälöihin (4.4) ja (4.5), jolloin saadaan yhtälöt

$$\begin{aligned} (\nu \nabla u^k, \nabla(u^k - u)) + (\mu u^k, u^k - u) - (p^k, \operatorname{div}(u^k - u)) &= (f, u^k - u) \\ (\nu \nabla u, \nabla(u^k - u)) + (\mu u, u^k - u) - (p, \operatorname{div}(u^k - u)) &= (f, u^k - u). \end{aligned}$$

Vähennetään nämä toisistaan, jolloin saadaan

$$(\nu \nabla(u^k - u), \nabla(u^k - u)) + (\mu(u^k - u), u^k - u) - (p^k - p, \operatorname{div}(u^k - u)) = 0.$$

Huomataan, että edellisen yhtälön vasemman puolen kaksi ensimmäistä termiä muodostavat funktion $u^k - u$ energianormin. Määritellään nyt $v^k := u^k - u$ ja $q^k := p^k - p$, jolloin yhtälö voidaan kirjoittaa muotoon

$$\|v^k\|^2 = (q^k, \operatorname{div} v^k). \quad (4.6)$$

Yhtälö (4.2) on yhtäpitävä vastaavan heikon muodon

$$(p^{k+1} - p^k, \phi) + \rho(\operatorname{div} u^k, \phi) = 0 \quad \forall \phi \in L_2(\Omega)$$

kanssa. Asettamalla $\phi = p^{k+1} - p$ saadaan

$$(p^{k+1} - p^k, p^{k+1} - p) + \rho(\operatorname{div} u^k, p^{k+1} - p) = 0,$$

joka edelleen yhtälön (4.6) edellä käyttöön otettujen merkintöjen avulla ilmaistuna on

$$(q^{k+1} - q^k, q^{k+1}) + \rho(\operatorname{div} v^k, q^{k+1}) = 0.$$

Toiseen korottamalla saadaan yhtäpitävästi

$$\|q^{k+1}\|^2 - \|q^k\|^2 + \|q^{k+1} - q^k\|^2 = -2\rho(\operatorname{div} v^k, q^{k+1}). \quad (4.7)$$

Kertomalla yhtälö (4.6) puolittain 2ρ :lla ja lisäämällä se yhtälöön (4.7), saadaan

$$\begin{aligned} \|q^{k+1}\|^2 - \|q^k\|^2 + \|q^{k+1} - q^k\|^2 + 2\rho \|v^k\|^2 &= -2\rho(\operatorname{div} v^k, q^{k+1} - q^k) \\ &\leq -2\rho \|\operatorname{div} v^k\| \|q^{k+1} - q^k\| \\ &\leq \frac{\rho^2}{\delta} (\|\nabla v^k\|^2 + \|v^k\|^2) + \delta \|q^{k+1} - q^k\|^2, \end{aligned} \quad (4.8)$$

missä $0 < \delta < 1$ on mielivaltainen vakio. Huomataan, että

$$\|v^k\|^2 \geq \underline{\nu} \|\nabla v^k\|^2 + \underline{\mu} \|v^k\|^2 \geq \min(\underline{\nu}, \underline{\mu}) (\|\nabla v^k\|^2 + \|v^k\|^2).$$

Siten

$$\begin{aligned} \|q^{k+1}\|^2 - \|q^k\|^2 + (1 - \delta) \|q^{k+1} - q^k\|^2 \\ + \rho \left(2 \min(\underline{\nu}, \underline{\mu}) - \frac{\rho}{\delta} \right) (\|\nabla v^k\|^2 + \|v^k\|^2) \leq 0. \end{aligned} \quad (4.9)$$

Jos nyt tehdään summaus $k = 0, \dots, N$, saadaan

$$\begin{aligned} & \|q^{N+1}\|^2 + (1 - \delta) \sum_{k=0}^N \|q^{k+1} - q^k\|^2 \\ & + \rho \left(2 \min(\underline{\nu}, \underline{\mu}) - \frac{\rho}{\delta} \right) \sum_{k=0}^N (\|\nabla v^k\|^2 + \|v^k\|^2) \leq \|q^0\|^2. \end{aligned} \quad (4.10)$$

Lauseen ehdon (4.3) nojalla on olemassa δ_* siten, että

$$0 < \frac{\rho}{2 \min(\underline{\nu}, \underline{\mu})} < \delta_* < 1$$

ja siten

$$2 \min(\underline{\nu}, \underline{\mu}) - \frac{\rho}{\delta_*} > 0.$$

Voidaan valita $\delta = \delta_*$, jolloin epäyhtälöstä (4.10) nähdään, että

$$\begin{aligned} \|q^{k+1} - q^k\|^2 &= \|p^{k+1} - p^k\|^2 \xrightarrow{k \rightarrow \infty} 0 \\ \|\nabla v^k\|^2 + \|v^k\|^2 &= \|\nabla(u^k - u)\|^2 + \|u^k - u\|^2 \xrightarrow{k \rightarrow \infty} 0, \end{aligned}$$

joten konvergenssi $u^k \xrightarrow{k \rightarrow \infty} u$ on osoitettu.

Edellisestä yhtälöstä nähdään myös, että p^k on rajoitettu $L_2(\Omega)$:ssa. Tällöin löydetään $L_2(\Omega)$:ssa heikosti suppeneva osajono $p^{k'} \rightarrow p_*$. Yhtälön (4.5) mukaan

$$(\nu \nabla u, \nabla v) + (\mu u, v) - (p_*, \operatorname{div} v) = (f, v) \quad \forall v \in V_0(\Omega)$$

ja vertaamalla tätä yhtälöön (4.4) saadaan, että

$$(p - p_*, \operatorname{div} v) = 0 \quad \forall v \in V_0,$$

jolloin

$$\nabla(p - p_*) = 0 \quad \Rightarrow \quad p_* = p + \text{vakio}.$$

Jokaisesta p^k :n osajonosta löytyy $L_2(\Omega)$:ssa heikosti konvergoiva osajono joka suppenee kohti $p + \text{vakio}$. Siksi jono p^k suppenee heikosti $\tilde{L}_2(\Omega)$:ssa kohti tarkkaa painekenttää p .

Huomautus 4.2.2. Määritetään painekenttä p asettamalla

$$\int_{\Omega} p(x) \, dx = 0$$

eli $p \in \tilde{L}_2(\Omega)$. Oletetaan, että $p^0 \in \tilde{L}_2(\Omega)$. Tällöin $p^k \in \tilde{L}_2(\Omega)$ kaikilla k , sillä

$$\begin{aligned} 0 &= \int_{\Omega} p^k(x) \, dx = \underbrace{\int_{\Omega} p^{k-1} \, dx}_{=0} - \rho \int_{\Omega} \operatorname{div} u^{k-1} \, dx \\ &= \int_{dS} u \cdot \hat{n} \, dS = 0, \end{aligned}$$

sillä u :n oletettiin häviävän Ω :n reunalla. Tässä \hat{n} on Ω :n reunan ulkonormaali. Näin ollen p^k konvergoi kohti p :tä avaruudessa $L_2(\Omega)$.

□

5 Funktionaaliset a posteriori virhearviot

Numeerisia likiarvomenetelmiä käytettäessä on tärkeää tietää, että kuinka hyviä saadut ratkaisut ovat, ts. kuinka lähelle tarkkaa ratkaisua numeerisella menetelmällä päästään. Tässä työssä virhearviot ovat Prof. Repinin kehittämät *funktionaaliset a posteriori virhearviot*.

Tarkemmin funktionaalisiin virhearvioihin liittyvää teoriaa on esitetty kirjoissa [19] ja [22]. Funktionaaliset virhearviot ovat sovellettavissa kaikille konformeille menetelmille, eivätkä arviot riipu käytetystä likiarvomenetelmästä. Tässä konformius tarkoittaa sitä, että numeerinen approksimaatio kuuluu samaan avaruuteen kuin tarkka ratkaisu, eli se toteuttaa kyseisen avaruuden sileyshdot. Virhearvioissa esiintyvät vakiot riippuvat vain käsiteltävästä alueesta Ω , ne eivät riipu käytetystä elementtiverkosta.

Tässä luvussa johdetaan funktionaaliset a posteriori virhearviot nopeuskentälle. Tämän tutkielman puitteissa ei tarkastella paineen virheen ylärajoja. Funktionaalisia virhearvioita Stokesin yhtälölle on johtanut Repin ja Stenberg [23]. Asiaa on myös tarkasteltu väitöskirjassa [8].

Samoja tekniikkoja käyttäen johdamme tässä gradussa funktionaaliset a posteriori virhearviot ottaen erityisesti huomioon Uzawa-algoritmin erityispiirteet. Vastaavia tuloksia on johdettu artikkelissa [1], jossa allekirjoittanut oli yksi kirjoittajista. Lisäksi artikkelissa käsitellään ns. augmentoitua Stokesin ongelmaa ja johdetaan sille vastaavat virhearviot.

5.1 Nopeuskentän virhemajorantti

Johdetaan aluksi yläraja divergenssivapaille approksimaatioille. Tämä yläraja on alunperin johdettu artikkelissa [23].

Lause 5.1.1. *Kun $v \in S_0(\Omega)$, kaikille $q \in \tilde{L}_2(\Omega)$, $\tau \in \Sigma(\Omega)$ ja $\beta \in \mathbb{R}_+$ pätee arvio*

$$\begin{aligned} \|u - v\|^2 &\leq \int_{\Omega} H(\beta, \mu) r^2(v, \tau) \, dx + \frac{1 + \beta}{\beta} \|\nu^{-1/2} d(v, \tau, q)\|^2 \\ &=: M_{\oplus}^{(1)}(v, \tau, q, \beta), \end{aligned}$$

missä

$$H(\beta, \mu) = \frac{C_\Omega^2(1 + \beta)}{C_\Omega^2\mu(1 + \beta) + \underline{\nu}}, \quad (5.1)$$

$$r(v, \tau) = f - \mu v + \text{Div}\tau, \quad (5.2)$$

$$d(v, \tau, q) = \tau - \nu \nabla v + \mathbb{1}q. \quad (5.3)$$

Tässä 1:lla merkitään yksikkötensoria.

Todistus. Lähtökohtana on luvussa 3 esitelty heikko muoto

$$\int_{\Omega} (\nu \nabla u : \nabla w + \mu u \cdot w) \, dx = \int_{\Omega} f \cdot w \, dx.$$

Lisäämällä heikkoon muotoon puolittain $\int_{\Omega} (\nu \nabla v : \nabla w + \mu v \cdot w) \, dx$, saadaan yhtälö

$$\begin{aligned} \int_{\Omega} (\nu \nabla(u - v) : \nabla w + \mu(u - v) \cdot w) \, dx \\ = \int_{\Omega} (f \cdot w - \nu \nabla v : \nabla w - \mu v \cdot w) \, dx. \end{aligned} \quad (5.4)$$

Oletetaan tunnetuksi osittaisintegroinnin kautta saatavat yhtälöt

$$\begin{aligned} \int_{\Omega} (\text{Div}\tau \cdot w + \tau : \nabla w) \, dx &= 0 \quad \forall \tau \in \Sigma(\text{Div}, \Omega), w \in V_0(\Omega), \\ \int_{\Omega} (\nabla q \cdot w + q \text{div}w) \, dx &= 0 \quad \forall q \in L_2(\Omega), w \in V_0(\Omega), \end{aligned}$$

ja kun valittiin $w \in S_0(\Omega)$, niin $\text{div}w = 0$. Näiden yhtälöiden avulla yhtälön (5.4) oikea puoli saadaan muotoon

$$\begin{aligned} \int_{\Omega} \left((f - \mu v + \text{Div}\tau) \cdot w + (\tau - \nu \nabla v) : \nabla w \right) \, dx \\ = \int_{\Omega} \frac{\alpha}{\mu^{1/2}} (f - \mu v + \text{Div}\tau) \cdot w \mu^{1/2} \, dx \\ + \int_{\Omega} \nu^{-1/2} (1 - \alpha) (f - \mu v + \text{Div}\tau) \cdot w \nu^{1/2} \, dx \\ + \int_{\Omega} \nu^{-1/2} (\tau - \nu \nabla v + \mathbb{1}q) : \nabla w \nu^{1/2} \, dx \\ = \int_{\Omega} \frac{\alpha}{\mu^{1/2}} r(v, \tau) \cdot w \mu^{1/2} \, dx + \int_{\Omega} \nu^{-1/2} (1 - \alpha) r(v, \tau) \cdot w \nu^{1/2} \, dx \\ + \int_{\Omega} \nu^{-1/2} d(v, \tau, q) : \nabla w \nu^{1/2} \, dx, \end{aligned}$$

missä $\alpha = \alpha(x)$ on funktio, joka saa arvoja väliltä $[0,1]$, $r(v, \tau)$ on määritelty yhtälössä (5.2) ja $d(v, \tau, q)$ yhtälössä (5.3). Lisäksi Hölderin (lemma 2.3.1) ja Friedrichin epäyhtälöiden 2.3.4 nojalla pätee

$$\begin{aligned} & \left| \int_{\Omega} r(v, \tau) \cdot w \, dx \right| \\ & \leq \left\| \frac{\alpha}{\mu^{1/2}} r(v, \tau) \right\| \|\mu^{1/2} w\| \\ & \quad + C_{\Omega} \underline{\nu}^{-1/2} \|(1 - \alpha)r(v, \tau)\| \|\nu^{1/2} \nabla w\|. \end{aligned}$$

Näin ollen saadaan

$$\begin{aligned} & \int_{\Omega} \left(\nu \nabla(u - v) : \nabla w + \mu(u - v) \cdot w \right) dx \\ & \leq \left(C_{\Omega} \underline{\nu}^{-1/2} \|\nu^{-1/2}(1 - \alpha)r(v, \tau)\| + \|\nu^{-1/2}d(v, \tau, q)\| \right) \|\nu^{1/2} \nabla w\| \\ & \quad + \left\| \frac{\alpha}{\mu^{1/2}} r(v, \tau) \right\| \|\mu^{1/2} w\|, \end{aligned}$$

josta Cauchy-Schwarzin epäyhtälön muotoa $ab + cd \leq \sqrt{a^2 + c^2} \sqrt{b^2 + d^2}$ käyttämällä saadaan edellisen yhtälön oikea puoli muotoon

$$\begin{aligned} & \left(\left(C_{\Omega} \underline{\nu}^{-1/2} \|(1 - \alpha)r(v, \tau)\| + \|\nu^{-1/2}d(v, \tau, q)\| \right)^2 \right. \\ & \quad \left. + \left\| \frac{\alpha}{\mu^{1/2}} r(v, \tau) \right\|^2 \right)^{1/2} \underbrace{\left(\|\mu^{1/2} w\|^2 + \|\nu^{1/2} \nabla w\|^2 \right)^{1/2}}_{=\|w\|}. \end{aligned}$$

Asetetaan nyt $w = u - v$, jolloin toiseen potenssiin korottamalla saamme virheelle ylärajan

$$\begin{aligned} \|u - v\|^2 & \leq \left(C_{\Omega} \underline{\nu}^{-1/2} \|(1 - \alpha)r(v, \tau)\| + \|\nu^{-1/2}d(v, \tau, q)\| \right)^2 \\ & \quad + \left\| \frac{\alpha}{\mu^{1/2}} r(v, \tau) \right\|^2. \end{aligned} \tag{5.5}$$

Tästä päästään erikoistapauksiin $\alpha = 0$ ja $\alpha = 1$:

$$\|u - v\| \leq C_\Omega \underline{\nu}^{-1/2} \|r(v, \tau)\| + \|\nu^{-1/2} d(v, \tau, q)\| =: M_0(v, \tau, q)$$

$$\|u - v\|^2 \leq \|\mu^{-1/2} r(v, \tau)\|^2 + \|\nu^{-1/2} d(v, \tau, q)\|^2 =: M_1(v, \tau, q).$$

Näistä $M_1(v, \tau, q)$ antaa tarkan ylärajan, kun $\tau = \sigma$ ja $q = p$, mutta pienillä μ :n arvoilla ensimmäisen termin kerroin $\mu^{-1/2}$ on suuri. Tämä tekee virhearviosta herkän residuaalin

$$r(v, \tau) := f - \mu v + \operatorname{Div} \tau$$

suhteen, ja tämä voi johtaa virheen yliarviointiin. $M_0(v, \tau, q)$:n kohdalla samaa ongelmaa ei ole, sen sijaan se ei välttämättä anna tarkkaa virhettä, eli ei tiedetä vastaako $\inf_{\tau, q} M_0(v, \tau, q)$ tarkan virheen energianormia. Näiden ongelmien välttämiseksi johdetaan funktiolle α optimaalinen muoto.

Olkoon $\beta > 0$ mielivaltainen. Kirjoitetaan yhtälö (5.5) Youngin epäyhtälön (Lemma 2.3.2) avulla muotoon

$$\begin{aligned} \|u - v\|^2 &\leq C_\Omega^2 (1 + \beta) \underline{\nu}^{-1} \|(1 - \alpha)r(v, \tau)\|^2 + \frac{1 + \beta}{\beta} \|\nu^{-1/2} d(v, \tau, q)\|^2 \\ &\quad + \left\| \frac{\alpha}{\mu^{1/2}} r(v, \tau) \right\|^2 \end{aligned}$$

Oikean puoleisen osan minimi α :n suhteen saavutetaan α :n arvolla

$$\alpha = \frac{C_\Omega^2 \mu (1 + \beta)}{C_\Omega^2 \mu (1 + \beta) + \underline{\nu}} \in [0, 1),$$

jolloin saadaan arvio

$$\|u - v\|^2 \leq \int_\Omega H(\beta, \mu) r^2(v, \tau) \, dx + \frac{1 + \beta}{\beta} \|\nu^{-1/2} d(v, \tau, q)\|^2 \quad (5.6)$$

missä $H(\beta, \mu)$ on määritelty yhtälössä (5.1). □

On helppoa nähdä, että johdettu majorantti on *tarkka*. Tämä tarkoittaa sitä, että teoriassa majorantti antaa tarkan virheen: valitsemalla $q = p$ ja $\tau = \nu \nabla u - \mathbb{1}p$ näemme, että

$$M_\oplus^{(1)}(v, \sigma, p, \beta) = \int_\Omega \underbrace{\frac{C_\Omega^2 (1 + \beta)}{C_\Omega^2 \mu (1 + \beta) + \underline{\nu}}}_{\xrightarrow{\beta \rightarrow \infty} \mu^{-1}} \mu^2 |u - v|^2 + \underbrace{\frac{1 + \beta}{\beta}}_{\xrightarrow{\beta \rightarrow \infty} 1} \nu |\nabla(u - v)|^2 \, dx,$$

joten

$$\inf_{\tau, q, \beta} M_{\oplus}^{(1)}(v, \tau, q, \beta) = \|u - v\|^2.$$

Koska divergenssivapaita approksimaatioita on vaikea laskea, johdamme hie-
man erilaisen ylärajan, jolla ei ole tätä vaatimusta. Myös tämä yläraja on
alunperin johdettu artikkelissa [23].

Lause 5.1.2. *Kun $\hat{v} \in V_0(\Omega)$, kaikille $q \in \tilde{L}_2(\Omega)$, $\tau \in \Sigma(\Omega)$ ja $\beta, \gamma, \delta \in \mathbb{R}_+$
pätee arvio*

$$\begin{aligned} \|u - \hat{v}\|^2 &\leq (1 + \gamma) \left(\int_{\Omega} (H(\beta, \mu)(1 + \delta)r^2(\hat{v}, \tau)) \, dx \right. \\ &\quad + (1 + \delta^{-1})\bar{c}C_{\Omega}^2\kappa_{\Omega}^2\|\operatorname{div}\hat{v}\|^2 \\ &\quad \left. + \frac{1 + \beta}{\beta}(\kappa_{\Omega}\bar{\nu}^{1/2}\|\operatorname{div}\hat{v}\| + \|\nu^{-1/2}d(\hat{v}, \tau, q)\|)^2 \right) \quad (5.7) \\ &\quad + (1 + \frac{1}{\gamma})\bar{C}_{\Omega}^2\kappa_{\Omega}^2\|\operatorname{div}\hat{v}\|^2 \\ &=: M_{\oplus}^{(2)}(\hat{v}, \tau, q, \beta, \gamma, \delta), \end{aligned}$$

missä

$$\begin{aligned} H(\beta, \mu) &= \frac{C_{\Omega}^2(1 + \beta)}{C_{\Omega}^2\mu(1 + \beta) + \underline{\nu}}, \\ r(\hat{v}, \tau) &= f - \mu\hat{v} + \operatorname{Div}\tau, \\ d(\hat{v}, \tau, q) &= \tau - \nu\nabla\hat{v} + \mathbb{1}q, \\ \bar{c} &= \sup_{x \in \Omega} H(\beta, \mu)\mu^2, \\ \bar{C}_{\Omega}^2 &= \sqrt{\bar{\mu}C_{\Omega}^2 + \bar{\nu}}. \end{aligned}$$

Todistus. Kaikille $w \in V_0(\Omega)$ pätee

$$\begin{aligned} \|w\|^2 &= \|\nu^{1/2}\nabla w\|^2 + \|\mu^{1/2}w\|^2 \\ &\leq \bar{\nu}\|\nabla w\|^2 + \bar{\mu}\|w\|^2 \\ &\leq \bar{\nu}\|\nabla w\|^2 + \bar{\mu}C_{\Omega}^2\|\nabla w\|^2 \\ &= (C_{\Omega}^2\bar{\mu} + \bar{\nu})\|\nabla w\|^2. \end{aligned} \quad (5.8)$$

Huomautuksen 2.3.6 nojalla löydetään funktio $v_0 \in S_0(\Omega)$ siten, että

$$\begin{aligned}\|\hat{v} - v_0\| &\leq \sqrt{C_\Omega^2 \bar{\mu} + \bar{\nu}} \|\nabla(\hat{v} - v_0)\| \\ &\leq \kappa_\Omega \sqrt{C_\Omega^2 \bar{\mu} + \bar{\nu}} \|\operatorname{div} \hat{v}\|,\end{aligned}$$

joten kolmioepäyhtälön ja Youngin epäyhtälön nojalla

$$\begin{aligned}\|u - \hat{v}\|^2 &\leq \left(\|u - v_0\| + \|v_0 - \hat{v}\| \right)^2 \\ &\leq (1 + \gamma) \|u - v_0\|^2 + \left(1 + \frac{1}{\gamma}\right) \|v_0 - \hat{v}\|^2 \\ &\leq (1 + \gamma) M_\oplus^{(1)}(v_0, \tau, q, \beta) + \left(1 + \frac{1}{\gamma}\right) \bar{C}_\Omega^2 \kappa_\Omega^2 \|\operatorname{div} \hat{v}\|^2,\end{aligned}$$

missä $\bar{C}_\Omega = \sqrt{\bar{\mu} C_\Omega^2 + \bar{\nu}}$ ja $\gamma > 0$.

Majorantti on nyt muotoa

$$M_\oplus^{(1)}(v_0, \tau, q, \beta) = \int_\Omega H(\beta, \mu) r^2(v_0, \tau) \, dx + \frac{1 + \beta}{\beta} \|\nu^{-1/2} d(v_0, \tau, q)\|^2,$$

missä

$$H(\beta, \mu) := \frac{C_\Omega^2(1 + \beta)}{C_\Omega^2 \mu(1 + \beta) + \underline{\nu}}.$$

Huomaa, että kolmioepäyhtälön avulla saadaan

$$\begin{aligned}\|\nu^{-1/2} d(v_0, \tau, q)\| &= \|\nu^{-1/2}(\tau - \nu \nabla v_0 + \mathbb{1}q)\| \\ &\leq \|\nu^{1/2} \nabla(v_0 - \hat{v})\| + \|\nu^{-1/2}(\tau - \nabla \hat{v} + \mathbb{1}q)\| \\ &\leq \kappa_\Omega \bar{\nu}^{1/2} \|\operatorname{div} \hat{v}\| + \|\nu^{-1/2} d(\hat{v}, \tau, q)\|\end{aligned}$$

ja Youngin epäyhtälön avulla

$$\begin{aligned}
\int_{\Omega} H(\beta, \mu) r^2(v_0, \tau) \, dx &\leq \int_{\Omega} H(\beta, \mu) (r(\hat{v}, \tau) + \mu|v_0 - \hat{v}|)^2 \, dx \\
&\leq \int_{\Omega} H(\beta, \mu) (1 + \delta) r^2(\hat{v}, \tau) \, dx \\
&\quad + \int_{\Omega} H(\beta, \mu) (1 + \delta^{-1}) |v_0 - \hat{v}|^2 \mu^2 \, dx \\
&\leq \int_{\Omega} H(\beta, \mu) (1 + \delta) r^2(\hat{v}, \tau) \, dx \\
&\quad + (1 + \delta^{-1}) \bar{c} C_{\Omega}^2 \kappa_{\Omega}^2 \|\operatorname{div} \hat{v}\|^2,
\end{aligned}$$

missä $\bar{c} = \sup_{x \in \Omega} H(\beta, \mu) \mu^2$.

Nyt huomataan, että

$$\begin{aligned}
M_{\oplus}^{(1)}(v_0, \tau, q, \beta) &\leq \int_{\Omega} H(\beta, \mu) (1 + \delta) r^2(\hat{v}, \tau) \, dx + (1 + \delta^{-1}) \bar{c} C_{\Omega}^2 \kappa_{\Omega}^2 \|\operatorname{div} \hat{v}\|^2 \\
&\quad + \frac{1 + \beta}{\beta} (\kappa_{\Omega} \bar{\nu}^{1/2} \|\operatorname{div} \hat{v}\| + \|\nu^{-1/2} d(\hat{v}, \tau, q)\|)^2,
\end{aligned}$$

josta edelleen päästään haluttuun tulokseen

$$\begin{aligned}
\|u - \hat{v}\|^2 &\leq (1 + \gamma) \left(\int_{\Omega} H(\beta, \mu) (1 + \delta) r^2(\hat{v}, \tau) \, dx \right. \\
&\quad + (1 + \delta^{-1}) \bar{c} C_{\Omega}^2 \kappa_{\Omega}^2 \|\operatorname{div} \hat{v}\|^2 \\
&\quad + \frac{1 + \beta}{\beta} (\kappa_{\Omega} \bar{\nu}^{1/2} \|\operatorname{div} \hat{v}\| + \|\nu^{-1/2} d(\hat{v}, \tau, q)\|)^2 \Big) \\
&\quad + (1 + \frac{1}{\gamma}) \bar{C}_{\Omega}^2 \kappa_{\Omega}^2 \|\operatorname{div} \hat{v}\|^2.
\end{aligned}$$

□

Tästä muodosta nähdään, että jos $\operatorname{div} \hat{v} = 0$, niin voidaan asettaa $\gamma = 0 = \delta$ ja $M_{\oplus}^{(2)}(\hat{v}, \tau, q, \beta, \gamma, \delta) = M_{\oplus}^{(1)}(\hat{v}, \tau, q, \beta)$.

5.2 Uzawa-algoritmin askelkohtainen virhe

Olkoon \mathcal{J}_h elementtiverkko, jonka tiheyttä kuvataan luvulla h . Olkoon $V_{0h}(\Omega, \mathbb{R}^n)$ ja $Q_h(\Omega)$ avaruuksien $V_0(\Omega, \mathbb{R}^n)$ ja $\tilde{L}_2(\Omega)$ äärellisulotteisia aliavaruuksia.

Oletetaan, että kaikille $v_h \in V_{0h}$ pätee $\operatorname{div} v_h \in Q_h$. Oletetaan myös, että avaruudet on rakennettu siten, että niille pätee ns. diskreetti LBB-ehto. Tämä tarkoittaa sitä, että kaikille $q_h \in Q_h$ on olemassa $v_h \in V_{0h}$ siten, että

$$\operatorname{div} v_h = q_h$$

ja

$$\|\nabla v_h\| \leq c\|q_h\|,$$

missä positiivinen vakio c ei riipu verkon tiheydestä h . Näillä oletuksilla diskreetti tehtävä on hyvin määritelty ja stabiili. Tämä tarkoittaa sitä, että ns. lukkoefektiä ei ilmene. Lisää stabiileista ja epästabiileista menetelmistä artikkelissa [24]. Kuitenkaan tässä tutkielmassa emme yksinkertaisuuden vuoksi käytä stabiilia avaruusparia. Avaruus V_{0h} muodostetaan ns. MINI-elementin avulla ja paineelle käytetään paloittain vakiota approksimaatiota.

Olkoon $u_h^k \in V_{0h}$ u^k :n approksimaatio, joka on laskettu verkossa \mathcal{J}_h . Seuraavaksi johdamme yhdistetyn virheen, joka tulee Uzawan algoritmista ja approksimaativirheestä.

Huomautuksen (2.3.6) avulla löydetään $u_0 \in S_0(\Omega)$ siten, että

$$\|\nabla(u^k - u_0)\| \leq \kappa_\Omega \|\operatorname{div} u^k\|.$$

Nyt Lauseen 3.3.1 ja yhtälön (5.8) nojalla saadaan

$$\begin{aligned} \|u - u^k\| &\stackrel{\Delta-ey}{\leq} \|u - u_0\| + \|u_0 - u^k\| \\ &\leq 2 \|u_0 - u^k\| \\ &\leq 2C \|\operatorname{div} u^k\|, \end{aligned} \tag{5.9}$$

missä

$$C = \kappa_\Omega \sqrt{C_\Omega^2 \bar{\mu} + \bar{\nu}}. \tag{5.10}$$

Voidaan siis tehdä arvio

$$\begin{aligned}
\|u - u_h^k\| &\stackrel{\Delta-ey}{\leq} \|u - u^k\| + \|u^k - u_h^k\| \\
&\leq 2C\|\operatorname{div}u^k\| + \|u^k - u_h^k\| \\
&\stackrel{\Delta-ey}{\leq} 2C\|\operatorname{div}u_h^k\| + 2C\|\operatorname{div}(u^k - u_h^k)\| + \|u^k - u_h^k\| \\
&\leq 2C\|\operatorname{div}u_h^k\| + 2C\|\nabla(u^k - u_h^k)\| + \|u^k - u_h^k\| \\
&\leq 2C\|\operatorname{div}u_h^k\| + 2C\underline{\nu}^{-1/2}\|\nu^{1/2}\nabla(u^k - u_h^k)\| + \|u^k - u_h^k\| \\
&\leq 2C\|\operatorname{div}u_h^k\| + (2C\underline{\nu}^{-1/2} + 1)\|u^k - u_h^k\|,
\end{aligned} \tag{5.11}$$

missä C on määritelty kuten edellä ja normi $\|\operatorname{div}u_h^k\|$ voidaan laskea. Ainoa tuntematon on siis $\|u^k - u_h^k\|$. Majorantti tälle johdetaan samaan tapaan, kuin majorantti virheelle $\|u - u^k\|$. Tämä yläraja on johdettu artikkelissa [1].

Lause 5.2.1. *Olkoon $u^k \in V_0(\Omega)$ Uzawa-algoritmin tarkka ratkaisu askeleessa k . Kun $u_h^k \in V_{0h}(\Omega)$ ratkaistaan Uzawa-algoritmin avulla, askelkohtainen virhe on*

$$\begin{aligned}
\|u^k - u_h^k\|^2 &\leq \int_{\Omega} H(\beta, \mu)r^2(u_h^k, \tau) \, dx + \frac{1 + \beta}{\beta}\|\nu^{-1/2}d(u_h^k, \tau, p^k)\|^2 \\
&=: M_{\oplus}^k(u_h^k, \tau, \beta, p^k) \quad \forall \tau \in \Sigma(\operatorname{Div}, \Omega), \forall \beta \in \mathbb{R}_+,
\end{aligned}$$

missä

$$\begin{aligned}
H(\beta, \mu) &= \frac{C_{\Omega}^2(1 + \beta)}{C_{\Omega}^2\mu(1 + \beta) + \underline{\nu}} \\
r(u_h^k, \tau) &= f - \mu u_h^k + \operatorname{Div}\tau \\
d(u_h^k, \tau, p^k) &= \tau - \nu \nabla u_h^k + \mathbb{1}p^k.
\end{aligned}$$

Todistus. Olkoon w funktio avaruudesta $V_0(\Omega)$. Tällöin yhtälöä (4.1) vastaa heikko muoto

$$\int_{\Omega} (\nu \nabla u^k : \nabla w + \mu u^k \cdot w - p^k \operatorname{div}w) \, dx = \int_{\Omega} f \cdot w \, dx. \tag{5.12}$$

Vähennetään yhtälöstä (5.12) puolittain $\int_{\Omega} (\nu \nabla v : \nabla w + \mu v \cdot w) \, dx$, jolloin saadaan yhtälö

$$\begin{aligned}
& \int_{\Omega} \left(\nu \nabla(u^k - v) : \nabla w + \mu(u^k - v) \cdot w \right) dx \\
&= \int_{\Omega} (f \cdot w - \nu \nabla v : \nabla w - \mu v \cdot w + p^k \operatorname{div} w) dx.
\end{aligned} \tag{5.13}$$

Osittaisintegroinnilla nähdään helposti, että

$$\int_{\Omega} (\operatorname{Div} \tau \cdot w + \tau : \nabla w) dx = 0 \quad \forall \tau \in \Sigma(\operatorname{Div}, \Omega), \quad w \in V_0(\Omega),$$

jonka avulla yhtälön (5.13) oikea puoli saadaan muotoon

$$\begin{aligned}
& \int_{\Omega} \left((f - \mu v + \operatorname{Div} \tau) \cdot w + (\tau - \nu \nabla v + \mathbb{1} p^k) : \nabla w \right) dx \\
&= \int_{\Omega} \frac{\alpha}{\mu^{1/2}} (f - \mu v + \operatorname{Div} \tau) \cdot w \mu^{1/2} dx \\
&\quad + \int_{\Omega} \nu^{-1/2} (1 - \alpha) (f - \mu v + \operatorname{Div} \tau) \cdot w \nu^{1/2} dx \\
&\quad + \int_{\Omega} \nu^{-1/2} (\tau - \nu \nabla v + \mathbb{1} p^k) : \nabla w \nu^{1/2} dx \\
&= \int_{\Omega} \frac{\alpha}{\mu^{1/2}} r(v, \tau) \cdot w \mu^{1/2} dx \\
&\quad + \int_{\Omega} \nu^{-1/2} (1 - \alpha) r(v, \tau) \cdot w \nu^{1/2} dx \\
&\quad + \int_{\Omega} \nu^{-1/2} d(v, \tau, p^k) : \nabla w \nu^{1/2} dx,
\end{aligned}$$

missä $\alpha = \alpha(x)$ on funktio, joka saa arvoja väliltä $[0,1]$. Lisäksi Hölderin (lemma 2.3.1) ja Friedrechin epäyhtälöiden (lemma 2.3.4) nojalla pätee

$$\begin{aligned}
& \left| \int_{\Omega} r(v, \tau) \cdot w dx \right| \\
&\leq \left\| \frac{\alpha}{\mu^{1/2}} r(v, \tau) \right\| \|\mu^{1/2} w\| \\
&\quad + C_{\Omega} \nu^{-1/2} \|(1 - \alpha) r(v, \tau)\| \|\nu^{1/2} \nabla w\|.
\end{aligned}$$

Näin ollen saadaan

$$\begin{aligned}
& \int_{\Omega} \left(\nu \nabla(u^k - v) : \nabla w + \mu(u^k - v) \cdot w \right) dx \\
& \leq \left(C_{\Omega} \underline{\nu}^{-1/2} \|(1 - \alpha)r(v, \tau)\| \right. \\
& \quad \left. + \|\nu^{-1/2}d(v, \tau, p^k)\| \right) \|\nu^{1/2}\nabla w\| \\
& \quad + \left\| \frac{\alpha}{\mu^{1/2}}r(v, \tau) \right\| \|\mu^{1/2}w\|,
\end{aligned}$$

josta Cauchy-Schwarzin epäyhtälön muodon $ab + cd \leq \sqrt{a^2 + c^2}\sqrt{b^2 + d^2}$ avulla edellisen yhtälön oikea puoli saadaan muotoon

$$\begin{aligned}
& \left((C_{\Omega} \underline{\nu}^{-1/2} \|(1 - \alpha)r(v, \tau)\| + \|\nu^{-1/2}d(v, \tau, p^k)\|)^2 \right. \\
& \quad \left. + \left\| \frac{\alpha}{\mu^{1/2}}r(v, \tau) \right\|^2 \right)^{1/2} \underbrace{\sqrt{\|\mu^{1/2}w\|^2 + \|\nu^{1/2}\nabla w\|^2}}_{=\|w\|}.
\end{aligned}$$

Asetetaan nyt $w = u^k - v$, jolloin toiseen potenssiin korottamalla saamme virheelle ylärajan

$$\begin{aligned}
\|u^k - v\|^2 & \leq \left(C_{\Omega} \underline{\nu}^{-1/2} \|(1 - \alpha)r(v, \tau)\| + \|\nu^{-1/2}d(v, \tau, p^k)\| \right)^2 \\
& \quad + \left\| \frac{\alpha}{\mu^{1/2}}r(v, \tau) \right\|^2. \tag{5.14}
\end{aligned}$$

Tästä päästään erikoistapauksiin $\alpha = 0$ ja $\alpha = 1$:

$$\|u^k - v\| \leq C_{\Omega} \underline{\nu}^{-1/2} \|r(v, \tau)\| + \|\nu^{-1/2}d(v, \tau, p^k)\| =: M_0^k(v, \tau, p^k)$$

$$\|u^k - v\|^2 \leq \|\mu^{-1/2}r(v, \tau)\|^2 + \|\nu^{-1/2}d(v, \tau, p^k)\|^2 =: M_1^k(v, \tau, p^k).$$

Näistä $M_1^k(v, \tau)$ antaa tarkan ylärajan, kun $\tau = \sigma$, mutta pienillä μ :n arvoilla ensimmäisen termin kerroin $\mu^{-1/2}$ on suuri. Tämä tekee virhearviosta herkän residuaalin

$$r(v, \tau) := f - \mu v + \text{Div} \tau$$

suhteen, ja tämä voi johtaa virheen yliarviointiin. $M_0^k(v, \tau)$:n kohdalla samaa ongelmaa ei ole, sen sijaan se ei välttämättä anna tarkkaa virhettä, eli ei tiedetä vastaako $\inf_{\tau} M_0^k(v, \tau)$ tarkan virheen energianormia. Kuten aiemmin, näiden ongelmien välttämiseksi johdetaan funktiolle α optimaalinen muoto.

Olkoon $\beta > 0$ mielivaltainen. Kirjoitetaan yhtälö (5.14) Youngin epäyhtälön (lemma 2.3.2) avulla muotoon

$$\begin{aligned} \|u^k - v\|^2 &\leq C_{\Omega}^2(1 + \beta)\|(1 - \alpha)r(v, \tau)\|^2 \\ &\quad + \frac{1 + \beta}{\beta}\|\nu^{-1/2}d(v, \tau, p^k)\|^2 \\ &\quad + \left\| \frac{\alpha}{\mu^{1/2}}r(v, \tau) \right\|^2. \end{aligned}$$

Oikean puoleisen osan minimi α :n suhteen saavutetaan α :n arvolla

$$\alpha = \frac{C_{\Omega}^2\mu(1 + \beta)}{C_{\Omega}^2\mu(1 + \beta) + \nu} \in [0, 1),$$

jolloin asettamalla $v = u_h^k$ saadaan haluttu arvio

$$\|u^k - u_h^k\|^2 \leq \int_{\Omega} H(\beta, \mu)r^2(u_h^k, \tau) \, dx + \frac{1 + \beta}{\beta}\|\nu^{-1/2}d(u_h^k, \tau, p^k)\|^2.$$

□

Yhtälön (5.11) ja lauseen 5.2.1 nojalla pätee

Lause 5.2.2. *Kun $u_h^k \in V_{0h}(\Omega)$ ratkaistaan Uzawa-algoritmin avulla, kaikille $\tau \in \Sigma(\text{Div}, \Omega)$ ja $\beta \in \mathbb{R}_+$ pätee arvio*

$$\begin{aligned} \|u - u_h^k\| &\leq 2C\|\text{div}u_h^k\| + (2C\nu^{-1/2} + 1)\sqrt{M_{\oplus}^k(u_h^k, \tau, \beta, p^k)} \\ &=: \mathcal{M}_{\oplus}(u_h^k, \tau, \beta, p^k) \end{aligned}$$

Huomautus 5.2.3. *Majorantissa esiintyvä funktio $p^k \in \tilde{L}_2(\Omega)$ on tarkassa ratkaisuparissa (u^k, p^k) oleva funktio. Jokaista $q \in \tilde{L}_2(\Omega)$ vastaa jokin funktio $v \in V_0$ siten, että heikko muoto (3.3) toteutuu [25]. Näin ollen yhtälö (5.9) pätee, vaikka p^k :n tilalla käytetään numeerista approksimaatiota $p_h^k \in Q_h$, sillä määritelmän mukaan $Q_h \subset \tilde{L}_2(\Omega)$. Ts. ed. lauseen majorantti on laskettavissa kun funktion p^k :n tilalla käytetään funktiota p_h^k .*

5.3 Majorantin globaali minimointi

Lauseen 5.2.2 ylärajassa esiintyy kaksi kaksi vapaata parametria, tesoriarvoinen funktio τ ja positiivinen reaaliluku β ja yläraja pätee kaikilla näiden arvoilla. Haluamme mahdollisimman tarkan ylärajan, joten M_{\oplus}^k tulee minimoida näiden molempien suhteen. Minimointi tehdään laskemalla Gateaux-derivaatta ylärajalle M_{\oplus}^k ja asettamalla se nolllaksi. Saadaan siis yhtälö

$$\begin{aligned} 0 &= \left. \frac{\partial M_{\oplus}^k(u_h^k, \tau + t\varphi, \beta, p^k)}{\partial t} \right|_{t=0} \\ &= \int_{\Omega} H(\beta, \mu)(f - \mu u_h^k + \text{Div}\tau) \cdot \text{Div}\varphi \, dx \\ &\quad + \frac{1 + \beta}{\beta} \int_{\Omega} \nu^{-1}(\tau - \nu \nabla u_h^k + \mathbb{1}p^k) : \varphi \, dx, \end{aligned}$$

kaikilla $\varphi \in \Sigma(\text{Div}, \Omega)$. Järjestelemällä termit uudelleen saadaan FEM-ongelma funktiolle τ :

$$\begin{aligned} \int_{\Omega} H(\beta, \mu) \text{Div}\tau \cdot \text{Div}\varphi \, dx + \frac{1 + \beta}{\beta} \int_{\Omega} \nu^{-1} \tau : \varphi \, dx \\ = - \int_{\Omega} H(\beta, \mu)(f - \mu u_h^k) \text{Div}\varphi \, dx \\ - \frac{1 + \beta}{\beta} \int_{\Omega} \nu^{-1}(-\nu \nabla u_h^k + \mathbb{1}p^k) : \varphi \, dx. \end{aligned} \tag{5.15}$$

Yhtälön (5.15) avulla löydetään siis optimaalinen τ , kun vakio β tunnetaan.

Minimointi etenee seuraavan rutiinin mukaisesti:

1. Annetaan β :lle alkuarvo 1 ja ratkaistaan τ yhtälön (5.15) avulla.
2. Minimoidaan β majorantin $M_{\oplus}^k(u_h^k, \tau, \beta)$ lausekkeesta laskemalla derivaatan nollakohta.
3. Ratkaistaan τ uudestaan vaiheessa 2 päivitettyllä β :n arvolla.
4. Mikäli lopetusehdot eivät täyty, siirrytään kohtaan 2.

6 Numeriikkaa

Tässä luvussa esitellään numeerisia tuloksia, jotka vahvistavat johdetun majorantin (lause 5.2.2) toimivuuden. Ennen numeerisia tuloksia tarkastellaan kirjallisuudesta löytyviä arvioita Friedrichin vakiolle C_Ω ja LBB-vakion käänteisluvulle κ_Ω ja esitellään lyhyesti elementtiratkaisijan kokoamisrutiini ja käytetyt elementit. Käytettyjen elementtiratkaisijoiden lähdekoodit ovat liitteenä (Liite A).

6.1 Vakioiden määrittely

Lemman 2.3.4 mukaan vakio C_Ω määriteltä

$$\|w\| \leq C_\Omega \|\nabla w\|.$$

Jos alueessa Ω on voimassa Dirichletin reunaehdot ja alue Ω voidaan upottaa neliöön, jonka sivun pituus on s , pätee [3, s. 30 ja 290] nojalla yhtälö

$$\|w\| \leq s\sqrt{2}\|\nabla w\|,$$

josta saadaan, että

$$C_\Omega \leq s\sqrt{2}.$$

Tutkittavassa tapauksessa on kuitenkin voimassa parempi arvio, joka on esitelty lemmän 2.3.4 yhteydessä.

Lemmassa 2.3.5 esiintyvälle vakiolle κ_Ω pätee

$$\kappa_\Omega = (C_{LBB})^{-1},$$

missä vakio C_{LBB} on peräisin ns. Ladyzhenskaya-Babuška-Brezzi-ehdosta. Selvästi nähdään, että pätee epäyhtälö $\kappa_\Omega \geq 1$.

Artikkeleiden [4] ja [6] nojalla C_{LBB} -vakiolle saadaan alueen Ω muodosta riippuen seuraavanlaisia arvioita: Jos $\Omega = (0, 1) \times (0, a)$, $a \geq 1$, niin

$$\frac{\sin \frac{\pi}{8}}{a} \leq C_{LBB} \leq \frac{\pi}{2\sqrt{3}a}.$$

Jos $\Omega = \{x = (x_1, x_2) : 0 < R_1 < |x| < R_2\}$, $\frac{R_2}{R_1} = 1 + \delta$, $0 < \delta \leq 1$, niin

$$C_{LBB} \leq \sqrt{\frac{7}{6}} \frac{\delta}{2}.$$

Jos $\Omega = \{(x_1, x_2) : 0 < x_i < L_i, i = 1, 2\}$ ja $l = \max\left(\frac{L_1}{L_2}, \frac{L_2}{L_1}\right)$, niin

$$\frac{1}{2\sqrt{15}l} \leq C_{LBB} \leq \frac{\pi}{2\sqrt{3}l}.$$

6.2 Elementtimenetelmä

Osittaisdifferentiaaliyhtälöiden analyttinen ratkaisu mielivaltaisen muotoisessa alueessa on yleensä työlästä ja vaikeaa, lähes poikkeuksetta jopa mahdotonta. Ratkaisuja voidaan etsiä erilaisilla likiarvomenetelmillä, kuten esimerkiksi tässä työssä käytetyllä elementtimenetelmällä (*eng. Finite Element Method, lyh. FEM*) [3, 5].

Elementtimenetelmä kehitettiin tarpeeseen ratkaista monimutkaisia elastikan ja rakenneanalyysin ongelmia. Ensimmäisen muodon siitä esittivät Alexander Hrennikoff ja Richard Courant 1940-luvun alkupuolella [20], nykyinen muoto taas on peräisin 1950-luvulta. Menetelmän saatua matemaattisen perustan, alkoivat myös matemaatikot osoittaa kiinnostusta sitä kohtaan.

Elementtimenetelmällä voidaan ratkaista osittaisdifferentiaaliyhtälöitä vaativissa tilanteissa, kuten monimutkaisessa tai muuttuvassa alueessa. Laskentaresursseja voidaan kohdistaa paikallisesti alueen eri osiin, jolloin säästetään laskenta-ajassa ja tarvittavassa laskentatehossa.

Elementtimenetelmässä on pohjimmiltaan kysymys funktioavaruuden approksimoinnista äärellisellä aliavaruudella, jonka kantafunktiot muodostetaan ns. elementtien avulla. Tasossa elementit ovat yleensä joko kolmion tai nelikulmion muotoisia. Kolmiulotteisia elementtejä käytettäessä käytetään vastavasti tetraedreja tai tiiliskivimäisiä elementtejä. Mallin geometriasta riippuen voidaan käyttää myös muunlaisia elementtejä. Saadun approksimaation tarkkuus riippuu elementtien määrästä, sekä niiden jakautumisesta alueessa.

Tässä työssä tulee ratkaistavaksi kaksi hieman toisistaan eroavaa FEM-ongelmaa. Ensimmäinen FEM-ongelma muodostuu jokaisella Uzawan algoritmin iteraatiolla, kun ratkaistaan Stokesin ongelma. Tässä FEM-ongelmassa käytetään ns. MINI-elementtiä [2] nopeuskentän approksimaation saamiseksi. Paineen approksimaatio muodostetaan projisoimalla nopeuskentän divergenssi paloittain vakioiden funktioiden joukkoon, siis paine on paloittain vakio. Näin valitut avaruudet eivät täytä diskreettiä LBB-ehtoa, joten on mahdollista, että esiintyy epästabiilisuutta. Tämän tutkielman puitteissa tämä ei kuitenkaan ole olennaista, sillä tavoitteena on tarkastella majorantin toimi-

vuutta.

Toinen FEM-ongelma muodostuu, kun etsitään majorantissa $M_{\oplus}^k(u_h^k, \tau, \beta, p^k)$ esiintyvää optimaalista tensoriarvoista funktiota τ , joka minimoi majorantin. Tämä FEM-ongelma ratkaistaan käyttämällä ns. Raviart-Thomas-elementtiä.

6.3 Osittaisdifferentiaaliyhtälön ratkaisijan muodostaminen

Tarkasteltavan ongelman heikko muoto on löytää $u \in V_0(\Omega)$ siten, että

$$\int_{\Omega} \nu \nabla u : \nabla w + \mu u \cdot w \, dx = \int_{\Omega} f \cdot w + p \operatorname{div} w \, dx \quad \forall w \in V_0(\Omega).$$

Kyseessä on ääretönulotteinen ongelma. Numeerista laskentaa varten ongelma tulee saattaa äärellisulotteiseen muotoon. Olkoon N_i jonkin $V_0(\Omega)$:n aliavaruuden kantafunktiot. Tällöin u :lle voidaan tehdä approksimaatio

$$u = \sum_{i=1}^n c_i N_i,$$

missä c_i :t ovat vakioarvoisia kertoimia. Nyt heikko muoto saadaan muotoon

$$\begin{aligned} \int_{\Omega} \nu \nabla \left(\sum_{i=1}^n c_i N_i \right) : \nabla N_j + \mu \left(\sum_{i=1}^n c_i N_i \right) \cdot N_j \, dx \\ = \int_{\Omega} f \cdot N_j + p \operatorname{div} N_j \, dx, \quad j = 1, \dots, n, \end{aligned}$$

joka on yhtäpitävä muodon

$$\sum_{i=1}^n c_i \int_{\Omega} (\nu \nabla N_i : \nabla N_j + \mu N_i \cdot N_j) \, dx = \int_{\Omega} f \cdot N_j + p \operatorname{div} N_j \, dx \quad j = 1, \dots, n$$

kanssa. Edelleen tästä muodosta saadaan yhtälöryhmä

$$Ac = b, \tag{6.1}$$

missä

$$\begin{aligned}
\{A_{ij}\} &= \int_{\Omega} (\nu \nabla N_i : \nabla N_j + \mu N_i \cdot N_j) \, dx \\
A &= \{A_{ij}\}_{i,j=1,\dots,n} \\
\{b_j\} &= \int_{\Omega} f \cdot N_j + p \operatorname{div} N_j \, dx \\
b &= \{b_j\}_{j=1,\dots,n} \\
c &= \{c_j\}_{j=1,\dots,n}.
\end{aligned}$$

Lineaarisessa yhtälöryhmässä (6.1) ainoa tuntematon on vektori c , joka voidaan ratkaista.

Yleensä kantafunktiot N_j valitaan siten, että ne eroavat nolasta vain rajoitetussa määrässä lähekkäisiä elementtejä. Tällöin integraalit A_{ij} ja b_j ovat nolaa suurimmassa osassa elementtejä. Näin globaaleista matriiseista tulee harvoja, joten yhtälöryhmän (6.1) ratkaisu on helpompaa ja nopeampaa.

Integrointi suoritetaan jokaiselle elementille erikseen ja lopuksi kootaan tulokset matriiseihin. Lisäksi numeerista integrointia ei tehdä itse elementissä, vaan hyödynnetään sopivaa affinikuvausta ja muuttujan vaihtoa, jotta integroinnit voidaan itseasiassa tehdä referenssielementissä, joksi on valittu yksikkökolmio tasossa. Tämä aiheuttaa myös sen, että kantafunktioita ja integrointipisteitä ei tarvitse muodostaa jokaiselle elementille erikseen, vaan ne voidaan kuvata aina referenssielementistä työn alla olevaan elementtiin.

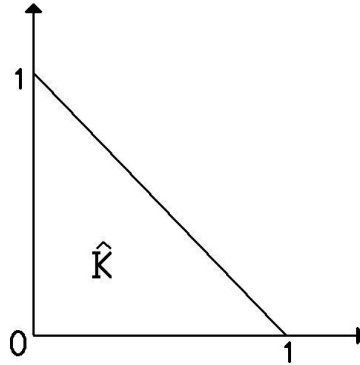
6.4 Affinikuvauksen muodostaminen

Affinikuvaus on kuvaus vektoriavaruudelta toiselle ja kuvaa pistejoukon toiseksi kiertäen, peilaten, skaalaten ja siirtäen. Affinikuvaus on muotoa

$$\begin{aligned}
F_K(\hat{x}) &= B_K \hat{x} + b_K \\
&= \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} \hat{x} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.
\end{aligned}$$

Jos vektori $b_K = 0$, niin kyseessä on puhtaasti lineaarikuvaus, jossa ei ole siirtoa.

Muodostetaan nyt kuvaus referenssielementiltä elementtiverkon elementille. Tässä työssä käytetään referenssielementtinä kuvan 2 mukaista yksikkökolmiota 2D:ssä.



Kuva 2: Referenssielementti.

Olkoon referenssielementin \hat{K} kulmapisteet, eli solmut \hat{p}_1 , \hat{p}_2 , \hat{p}_3 ja vastaavasti verkon elementin kulmapisteet p_1 , p_2 , p_3 . Muodostettavalta kuvaukselta halutaan, että

$$\begin{aligned} F_K(\hat{p}_1) &= p_1, \\ F_K(\hat{p}_2) &= p_2, \\ F_K(\hat{p}_3) &= p_3. \end{aligned}$$

Asettamalla $\hat{p}_1 = (0, 0)$, $\hat{p}_2 = (1, 0)$ ja $\hat{p}_3 = (0, 1)$ ja ratkaisemalla yllä oleva yhtälöryhmä saadaan

$$F_K(\hat{x}) = \begin{pmatrix} (p_2 - p_1)_x & (p_3 - p_1)_x \\ (p_2 - p_1)_y & (p_3 - p_1)_y \end{pmatrix} \hat{x} + \begin{pmatrix} p_{1x} \\ p_{1y} \end{pmatrix}.$$

6.5 Kantafunktioiden muodostaminen

Tässä osiossa johdetaan kantafunktiot MINI- ja Raviart-Thomas -elementeille. MINI-elementistä lisää artikkelissa [2] ja Raviart-Thomas -elementistä artikkelissa [21].

6.5.1 MINI-elementti

Muodostetaan aluksi skalaariarvoiset kantafunktiot $\hat{\phi}_j$ ja näistä edelleen tarvittavat vektoriarvoiset kantafunktiot. Tässä työssä käytetään kolmea lineaarista skalaariarvoista kantafunktiota ja neljäntenä niiden tulona saatavaa korkeampiasteista ns. kuplafunktiota. Lineaariset kantafunktiot ovat muotoa

$$\hat{\phi}_j(\hat{x}, \hat{y}) = a_j + b_j \hat{x} + c_j \hat{y}.$$

Asetetaan kantafunktiolle ϕ_j ehto

$$Z_{\hat{K}} = \{l_i, 1 \leq i \leq 3 : l_1(\phi_j) = \phi_j(0,0), l_2(\phi_j) = \phi_j(1,0), l_3(\phi_j) = \phi_j(0,1)\},$$

josta saadaan yhtälöryhmä

$$\begin{aligned} l_1(\phi_j) &= a_j \\ l_2(\phi_j) &= a_j + b_j \\ l_3(\phi_j) &= a_j + c_j. \end{aligned}$$

Tästä voidaan ratkaista kantafunktiot:

$$\begin{aligned} \phi_1 : \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} a_1 = 1 \\ b_1 = -1 \\ c_1 = -1 \end{cases} \\ \phi_2 : \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix} &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} a_2 = 0 \\ b_2 = 1 \\ c_2 = 0 \end{cases} \\ \phi_3 : \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_3 \\ b_3 \\ c_3 \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \Rightarrow \begin{cases} a_3 = 0 \\ b_3 = 0 \\ c_3 = 1. \end{cases} \end{aligned}$$

Näin on saatu johdettua kolme ensimmäistä kantafunktiota

$$\begin{aligned} \hat{\phi}_1(\hat{x}, \hat{y}) &= 1 - \hat{x} - \hat{y}, \\ \hat{\phi}_2(\hat{x}, \hat{y}) &= \hat{x}, \\ \hat{\phi}_3(\hat{x}, \hat{y}) &= \hat{y}, \end{aligned}$$

edelleen saadaan neljäs kantafunktio

$$\hat{\phi}_4(\hat{x}, \hat{y}) = \hat{x}\hat{y}(1 - \hat{x} - \hat{y}).$$

Nämä neljä kantafunktiota muodostavat skalaariarvoisen MINI-elementin [2]. Tätä elementtiä voidaan myös käyttää vektoriarvoiselle FEM-tehtävälle "monistamalla" skalaariarvoinen MINI-elementti jokaiselle komponentille:

$$\begin{aligned}
\hat{N}_1(\hat{x}, \hat{y}) &= (\hat{\phi}_1(\hat{x}, \hat{y}), 0) \\
\hat{N}_2(\hat{x}, \hat{y}) &= (0, \hat{\phi}_1(\hat{x}, \hat{y})) \\
\hat{N}_3(\hat{x}, \hat{y}) &= (\hat{\phi}_2(\hat{x}, \hat{y}), 0) \\
\hat{N}_4(\hat{x}, \hat{y}) &= (0, \hat{\phi}_2(\hat{x}, \hat{y})) \\
\hat{N}_5(\hat{x}, \hat{y}) &= (\hat{\phi}_3(\hat{x}, \hat{y}), 0) \\
\hat{N}_6(\hat{x}, \hat{y}) &= (0, \hat{\phi}_3(\hat{x}, \hat{y})) \\
\hat{N}_7(\hat{x}, \hat{y}) &= (\hat{\phi}_4(\hat{x}, \hat{y}), 0) \\
\hat{N}_8(\hat{x}, \hat{y}) &= (0, \hat{\phi}_4(\hat{x}, \hat{y})).
\end{aligned}$$

Kantafunktiot \hat{N}_j , $j = 1, \dots, 8$, muodostavat vektoriaarvoisen MINI-elementin. MINI-elementtiä on tarkasteltu yksityiskohtaisemmin artikkelissa [2]. MINI-elementin skalaariarvoiset kantafunktiot transformoidaan käsiteltävään elementtiin yhtälöiden

$$\begin{aligned}
\phi_j &= |\det(B_K)| \hat{\phi}_j \\
\nabla \phi_j &= |\det(B_K)| (B_K^{-1})^T \nabla \hat{\phi}_j
\end{aligned}$$

avulla. Tässä B_K on johdettu luvussa 6.4. Vektoriaarvoiset kantafunktiot transformoidaan komponentteittain.

6.5.2 Raviart-Thomas-elementti

Muodostetaan ensin vektoriaarvoiset kantafunktiot $\hat{\varphi}_j$ ja näistä edelleen tarvittavat tensoriarvoiset kantafunktiot. Tässä työssä käytetään tensoriarvoiselle ongelmalle lineaarista Raviart-Thomas-elementtiä, joten kantafunktiot ovat lineaarisia vektoriaarvoisia funktioita ja niitä on yksi jokaista elementin reunaa kohti. Kantafunktiot ovat

$$\begin{aligned}
\hat{\varphi}_1(\hat{x}, \hat{y}) &= \begin{pmatrix} \hat{x} \\ \hat{y} - 1 \end{pmatrix} \\
\hat{\varphi}_2(\hat{x}, \hat{y}) &= \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \\
\hat{\varphi}_3(\hat{x}, \hat{y}) &= \begin{pmatrix} \hat{x} - 1 \\ \hat{y} \end{pmatrix}.
\end{aligned}$$

Nämä voidaan edelleen tensoriarvoistaa asettamalla

$$\hat{M}_1(\hat{x}, \hat{y}) = \begin{pmatrix} \hat{x} & \hat{y} - 1 \\ 0 & 0 \end{pmatrix}$$

$$\hat{M}_2(\hat{x}, \hat{y}) = \begin{pmatrix} 0 & 0 \\ \hat{x} & \hat{y} - 1 \end{pmatrix}$$

$$\hat{M}_3(\hat{x}, \hat{y}) = \begin{pmatrix} \hat{x} & \hat{y} \\ 0 & 0 \end{pmatrix}$$

$$\hat{M}_4(\hat{x}, \hat{y}) = \begin{pmatrix} 0 & 0 \\ \hat{x} & \hat{y} \end{pmatrix}$$

$$\hat{M}_5(\hat{x}, \hat{y}) = \begin{pmatrix} \hat{x} - 1 & \hat{y} \\ 0 & 0 \end{pmatrix}$$

$$\hat{M}_6(\hat{x}, \hat{y}) = \begin{pmatrix} 0 & 0 \\ \hat{x} - 1 & \hat{y} \end{pmatrix}.$$

Nämä kantafunktiot muodostavat Raviart-Thomas-elementin [21]. Raviart-Thomas-elementtien yhteydessä on tärkeää tietää mihin suuntaan elementin normaalikomponentti (ulkonormaali) osoittaa. Jokaiseen reunaan liittyy kaksi elementtiä, joilla on luonnollisesti yhtäsuuret, mutta vastakkaisuuntaiset ulkonormaalit. Lisäksi kuvatessa kantafunktioita referenssielementiltä elementtiverkolle tulee normaalikomponenttien säilyä jatkuvina. Tämän vuoksi nämä kuvaukset tehdän Piola-muunnosta [9] hyödyntäen. Piola-muunnos ei kuitenkaan automaattisesti osaa huolehtia suunnistuksesta, vaan se pitää huomioida erikseen.

Vektoriarvoiset kantafunktiot Piola-muunnetaan seuraavien yhtälöiden mukaisesti:

$$\varphi_j = \frac{1}{|\det(B_K)|} B_K \text{sign}(\hat{\varphi}_j) \hat{\varphi}_j$$

$$\text{Div} \varphi_j = \frac{1}{|\det(B_K)|} \text{sign}(\hat{\varphi}_j) \text{Div} \hat{\varphi}_j,$$

missä $\text{sign}(\hat{\varphi}_j)$ on vektoriarvoisen kantafunktion $\hat{\varphi}_j$ suunnistus ja B_K on johdettu luvussa 6.4. Tensoriarvoiset kantafunktiot Piola-muunnetaan komponentteittain.

6.6 Numeeriset tulokset

Seuraavaksi tarkastelemme, että miten hyvin majorantti (lause 5.2.2) toimii käytännössä. Ensin tarkastellaan tilannetta, jossa tarkka ratkaisu tunnetaan ja todetaan majorantin pysyvän tarkan ratkaisun yläpuolella niin tiheällä kuin harvallaakin verkolla. Tämän jälkeen tarkastellaan tilannetta, jossa tarkka ratkaisu ei ole tiedossa ja samalla tarkastellaan Uzawa-algoritmissa esiintyvän parametrin ρ vaikutusta.

6.6.1 Tarkka ratkaisu tunnetaan

Tässä testiesimerkissä tarkka ratkaisu on tiedossa, joten voidaan vahvistaa, että johtamamme majorantti todella pysyy energianormissa mitatun tarkan virheen yläpuolella. Olkoon $\Omega = [0, 1] \times [0, 1]$ ja asetetaan parametreille ν ja μ vakioarvot 1. Yleistetyn Stokesin ongelman tarkka ratkaisu on

$$\begin{aligned} u(x, y) &= \begin{pmatrix} 20x^2y(2y-1)(x-1)^2(y-1) \\ -20xy^2(2x-1)(x-1)(y-1)^2 \end{pmatrix} \\ p(x, y) &= 2x - 1, \end{aligned}$$

joista voidaan edelleen laskea lastivektori f .

Luodaan epäsäännöllinen verkko, jossa on 1102 elementtiä. Asetetaan Uzawan parametriksi $\rho = 0,5$ ($< 2\min(\underline{\nu}, \underline{\mu}) = 2$). Tällöin tekemällä 20 Uzawa-algoritmin iteraatioaskelta saadaan taulukossa 1 ja kuvassa 3 esitetyt tulokset.

Tulosten perusteella nähdään, että majorantti todella pysyy tarkan virheen yläpuolella kaikilla Uzawa-algoritmin iteraatioilla. Huomionarvoista on myös se, että virheessä tehdään huomattavaa yliarviointia, joka johtuu suurimmaksi osaksi vakiosta κ_Ω . Yksikköneliön tapauksessa vakiolle κ_Ω saadaan edellä esitetty arvio

$$\frac{2\sqrt{3}}{\pi} \leq \kappa_\Omega \leq \frac{1}{\sin \frac{\pi}{8}}.$$

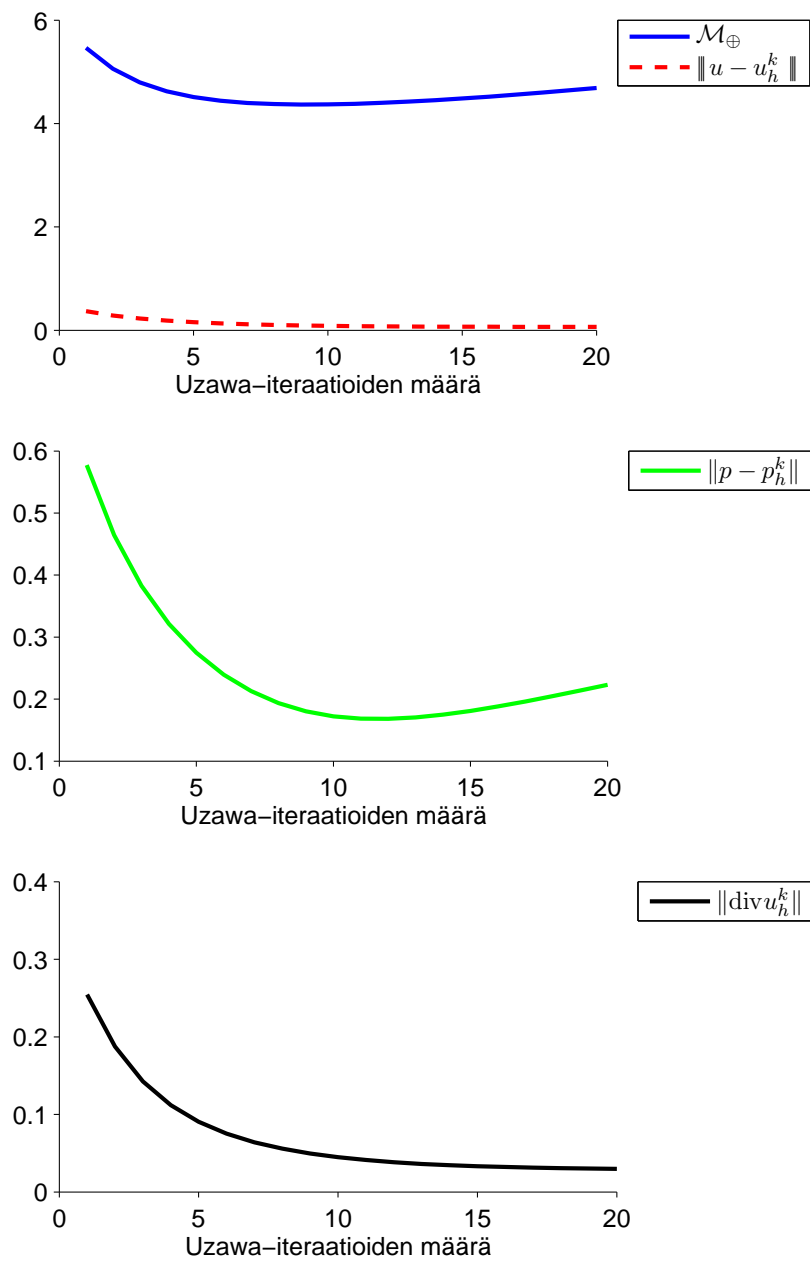
Tässä κ_Ω :lle on käytetty ylärajan arvoa $\frac{1}{\sin \frac{\pi}{8}}$. Pienemmillä κ_Ω :n arvoilla majorantin arvokin on luonnollisesti pienempi. Laskemalla vapaa parametri τ tiheämmässä verkossa kuin nopeuden likiarvo u_h^k saatettaisiin myös saada virhettä optimistisemmaksi. Tämän tyyliä testejä ei tämän tutkielman puitteissa tehdä.

Taulukko 1: Numeerisia tuloksia tiheämmällä verkolla.

Iter	$\ u - u_h^k\ $	$\frac{\mathcal{M}_\oplus}{\ u - u_h^k\ }$	\mathcal{M}_\oplus	$\ \operatorname{div}u_h^k\ $	$\sqrt{M_\oplus^k}$	$\ p - p_h^k\ $
1	0.3696	14.7881	5.4655	0.2545	0.64528	0.5774
2	0.2857	17.7151	5.0604	0.1875	0.63805	0.4646
3	0.2276	21.0751	4.7972	0.1426	0.63449	0.3825
4	0.1864	24.8085	4.6254	0.1119	0.63331	0.3213
5	0.1565	28.8467	4.5138	0.0905	0.63378	0.2748
6	0.1342	33.1099	4.4426	0.0752	0.63545	0.2395
7	0.1173	37.5033	4.3994	0.0641	0.63804	0.2129
8	0.1044	41.9161	4.3762	0.0558	0.64135	0.1936
9	0.0945	46.2275	4.3677	0.0496	0.64527	0.1803
10	0.0869	50.3186	4.3705	0.0449	0.64970	0.1722
11	0.0810	54.0880	4.3819	0.0412	0.65457	0.1684
12	0.0766	57.4636	4.4003	0.0384	0.65983	0.1681
13	0.0732	60.4094	4.4243	0.0362	0.66543	0.1705
14	0.0708	62.9238	4.4529	0.0346	0.67136	0.1750
15	0.0690	65.0326	4.4853	0.0332	0.67757	0.1810
16	0.0677	66.7790	4.5210	0.0322	0.68404	0.1882
17	0.0668	68.2146	4.5593	0.0314	0.69077	0.1961
18	0.0663	69.3920	4.6001	0.0307	0.69772	0.2047
19	0.0660	70.3605	4.6429	0.0302	0.70488	0.2137
20	0.0659	71.1630	4.6874	0.0298	0.71224	0.2230

Tuloksista nähdään myös, että majorantin arvot eivät laske jokaisella iteraatiolla. Tämä käytös on hyvin luonnollista, sillä paineen arviointi elementteittäin vakiolla funktiolla ei ole kovin tarkka. Tämän vuoksi paineen approksimaation virhe alkaa kasvaa alun iteraatioiden jälkeen (iteraatiossa 13), ks kuva 3 ja taulukko 1. Tätä epätarkkaa approksimaatiota käytetään myös seuraavan nopeuden approksimaation laskemiseen, joten tehty virhe kertautuu siellä, sillä tätä nopeuden aproksimaatiota käytetään edelleen seuraavan paineen laskemiseen. Tämä voi johtua myös siitä, että valittu äärellisulotteinen avaruuspari nopeuskentälle ja paineelle on epästabili, kuten aiemmin on todettu.

Tehokkuusindeksin $\frac{\mathcal{M}_\oplus}{\|u - u_h^k\|}$ havaitaan kasvavan jokaisella iteraatiolla. Tämä on seurausta siitä, että nopeuden tarkka virhe laskee majoranttia nopeammin. Uzawa-algoritmin havaitaan tekevän juuri sitä, mitä haluttiinkin, eli pienentävän approksimaation divergenssiä jokaisella iteraatiolla.



Kuva 3: Numeerisia tuloksia tiheämmällä verkolla.

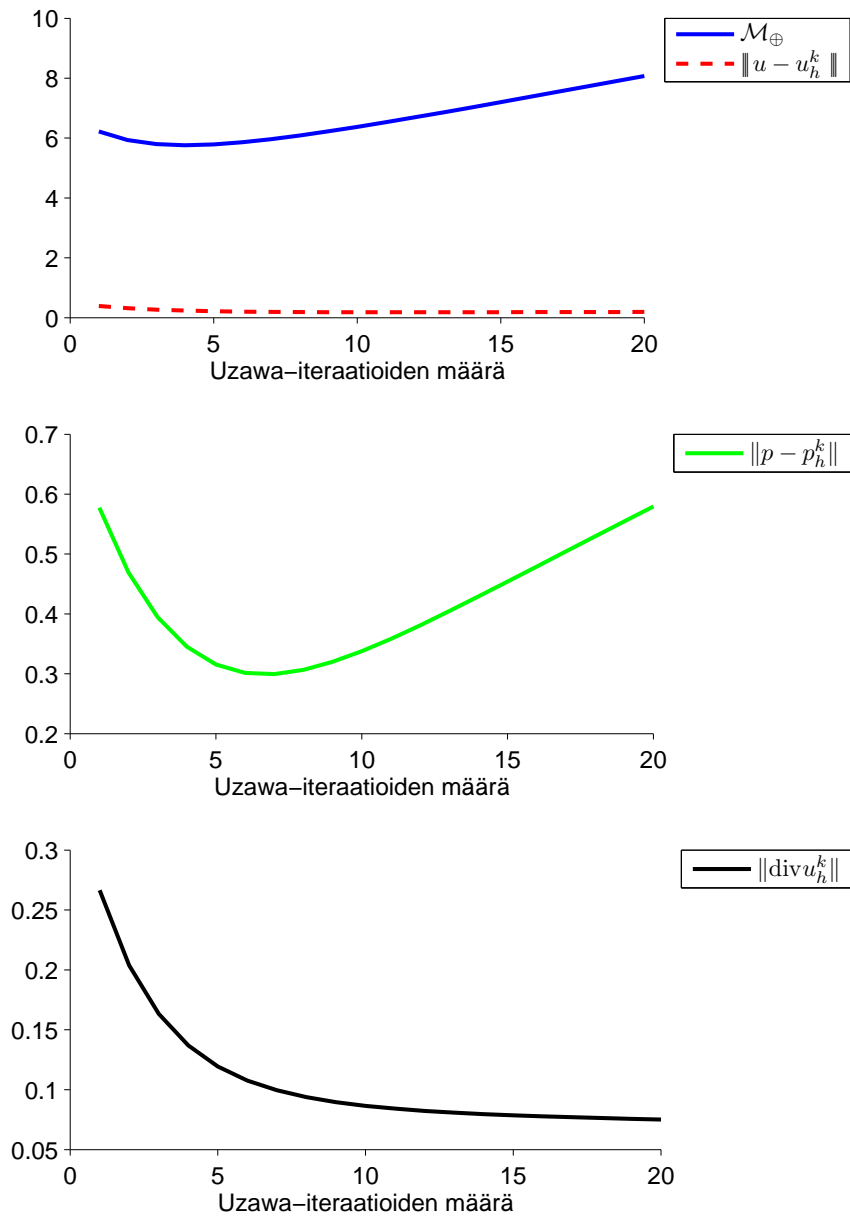
Tämä käytös ei ole riippuvaista verkon tiheydestä. Kun toistamme testin epä-säännöllisellä verkolla, jossa on vain 148 elementtiä, saamme vastaavanlaisia tuloksia, jotka on esitetty taulukossa 2 ja kuvassa 4.

Taulukko 2: Numeerisia tuloksia harvemmallalla verkolla.

Iter	$\ u - u_h^k\ $	$\frac{\mathcal{M}_\oplus}{\ u - u_h^k\ }$	\mathcal{M}_\oplus	$\ \operatorname{div}u_h^k\ $	$\sqrt{M_\oplus^k}$	$\ p - p_h^k\ $
1	0.3930	15.8277	6.2200	0.2665	0.75383	0.5774
2	0.3177	18.6721	5.9321	0.2038	0.76139	0.4690
3	0.2688	21.5676	5.7965	0.1633	0.77421	0.3945
4	0.2368	24.3230	5.7601	0.1369	0.79077	0.3452
5	0.2159	26.8106	5.7892	0.1194	0.81007	0.3156
6	0.2023	28.9772	5.8622	0.1076	0.83146	0.3015
7	0.1935	30.8268	5.9648	0.0995	0.85443	0.2995
8	0.1879	32.3954	6.0881	0.0938	0.87865	0.3065
9	0.1846	33.7303	6.2260	0.0896	0.90384	0.3199
10	0.1828	34.8782	6.3744	0.0865	0.92980	0.3376
11	0.1820	35.8799	6.5304	0.0842	0.95634	0.3583
12	0.1820	36.7684	6.6922	0.0823	0.98336	0.3809
13	0.1825	37.5691	6.8582	0.0808	1.0107	0.4046
14	0.1835	38.3018	7.0274	0.0796	1.0384	0.4291
15	0.1847	38.9811	7.1990	0.0786	1.0662	0.4540
16	0.1861	39.6177	7.3723	0.0777	1.0942	0.4791
17	0.1876	40.2198	7.5469	0.0770	1.1223	0.5043
18	0.1893	40.7933	7.7223	0.0763	1.1505	0.5294
19	0.1910	41.3426	7.8983	0.0757	1.1787	0.5545
20	0.1928	41.8710	8.0745	0.0751	1.2069	0.5794

Harvemman verkon tapauksessa tarkan virheen $\|u - u_h^k\|$ ja majorantin \mathcal{M}_\oplus arvot ovat luonnollisesti hieman suurempia kuin tiheämmällä verkolla. Suhteellisesti ajateltuna tarkassa virheessä tapahtuu suurempi kasvu, josta edelleen seuraa pienempi tehokkuusindeksi. Tässä tapauksessa paineen virheen havaitaan lähtevän kasvuun jo iteraation 8 kohdalla, ks kuva 4 ja taulukko 2. Tämän ilmiön aikaistuminen johtuu elementtien koon kasvamisesta. Edellä mainittu epätarkkuus paineen approksimointiin on suurempaa suuremmilla elementeillä.

Harvemmallalla verkolla havaitaan myös, että nopeuden tarkka virhe lähtee kasvuun iteraation 13 kohdalla. Pienempien elementtien antaessa tarkemman approksimaation, tätä nopeuden virheen kasvua ei ole havaittavissa tiheämmän verkon tapauksessa. Kuten tiheämmällä verkolla, avaruusparin epästabiilius voi olla syynä tähän ilmiöön myös harvan verkon tapauksessa.



Kuva 4: Numeerisia tuloksia harvemmalla verkolla.

6.6.2 Vapaiden parametrien minimointi

Tarkastellaan seuraavaksi sitä, mitä tapahtuu yhden Uzawa-algoritmin iteraation sisällä, eli minimoidessa parametreja β ja τ . Otetaan uudelleen tarkasteluun ensimmäisenä käsitelty esimerkki ja tarkastellaan Uzawa-algoritmien iteraatioita 1, 5, 10 ja 20 hieman tarkemmin. Jokaisella Uzawan iteraatiolla asetetaan aluksi $\beta = 1$. Näiden askelten sisällä tapahtuvat minimoinnit on koottu taulukkoon 3.

Taulukko 3: τ :n ja β :n minimointi.

Uzawan iter	Minimointi iter	β	\mathcal{M}_\oplus
1	1	1	5.9669
	2	1.6712	5.7418
	3	2.7073	5.5923
	4	4.1839	5.5061
	5	5.9808	5.4655
5	1	1	5.0721
	2	1.7746	4.8031
	3	3.0357	4.6354
	4	4.8819	4.5482
	5	7.0317	4.5138
10	1	1	5.0128
	2	1.9053	4.6781
	3	3.0357	4.4880
	4	5.7474	4.3998
	5	8.2941	4.3705
20	1	1	5.5680
	2	2.2686	5.0353
	3	4.7027	4.7919
	4	8.4971	4.7060
	5	12.1232	4.6874

Taulukon perusteella nähdään, että Uzawa-askeleen sisällä tapahtuvalla minimointi-iteraatiolla majorantin \mathcal{M}_\oplus arvo laskee. Tämä vahvistaa sen, että vapaiden parametrien β ja τ minimointi tapahtuu halutulla tavalla. Jokaisella iteraatiolla majorantin arvo muuttuu vähemmän, eli se alkaa vakioitua kohti jotakin arvoa.

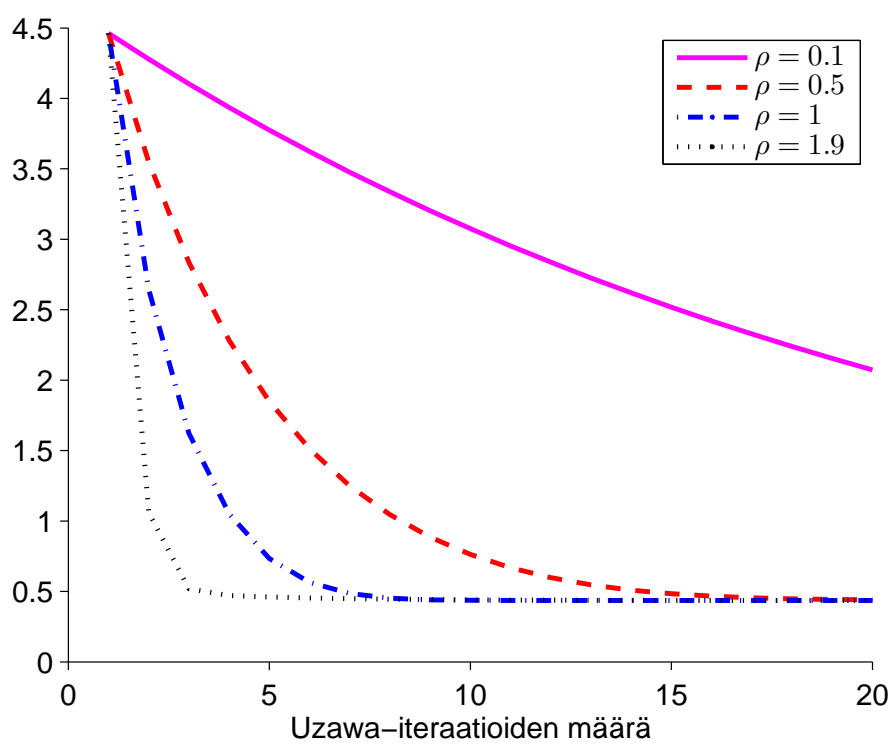
6.6.3 Tarkkaa ratkaisua ei tunneta

Tarkastellaan nyt tilannetta, jossa tarkkaa ratkaisua ei tunneta. Olkoot Ω yksikköympyrä ja $\nu = \mu = 1$. Asetetaan nyt lastifunktioksi vakioarvoinen vektorifunktio $f(x, y) = (1; 1)$. Tällöin Uzawa-algoritmin parametrin ρ arvoa vaihtelemalla saadaan majorantille \mathcal{M}_{\oplus} taulukon 4 ja kuvan 5 mukaiset arvot.

Taulukko 4: Majorantin arvoja eri Uzawa-algoritmin parametrin arvoilla.

Iteraatio	$\rho = 0.1$	$\rho = 0.5$	$\rho = 1$	$\rho = 1.9$
1	4.4634	4.4634	4.4634	4.4634
2	4.2794	3.5473	2.6419	1.0594
3	4.1037	2.8355	1.6231	0.5188
4	3.9358	2.2811	1.0512	0.4704
5	3.7756	1.8489	0.7334	0.4605
6	3.6225	1.5118	0.5648	0.4539
7	3.4762	1.2490	0.4851	0.4488
8	3.3365	1.0444	0.4525	0.4452
9	3.2030	0.8854	0.4408	0.4425
10	3.0755	0.7626	0.4370	0.4406
11	2.9536	0.6687	0.4359	0.4391
12	2.8371	0.5981	0.4356	0.4382
13	2.7258	0.5462	0.4356	0.4374
14	2.6195	0.5092	0.4356	0.4369
15	2.5178	0.4834	0.4356	0.4365
16	2.4206	0.4661	0.4356	0.4362
17	2.3278	0.4546	0.4356	0.4360
18	2.2390	0.4472	0.4356	0.4359
19	2.1541	0.4426	0.4356	0.4357
20	2.0730	0.4397	0.4356	0.4357

Tuloksista nähdään, että majorantti konvergoi nopeammin suuremmilla ρ :n arvoilla. Iteraatioiden edetessä majorantin arvojen havaitaan konvergoivan kohti samaa arvoa. Tämä voidaan todeta ainakin kolmen suurimman ρ :n arvon perustella. Pienimmällä ρ :n arvolla iteraatioita tulisi tehdä lisää tämän toteamiseksi, sillä konvergenssi on hyvin hidasta.



Kuva 5: Majorantin arvoja eri Uzawa-algoritmin parametrin arvoilla.

7 Johtopäätökset

Tämän tutkielman tarkoituksena oli tutkia yleistetyille Stokesin yhtälölle johdettuja a posteriori virhearvioita. Numeerisena menetelmänä käytettiin elementtimenetelmää ja approksimaatiomenetelmänä Uzawa-algoritmia. Elementtimenetelmän yhteydessä käytimme MINI-elementtiä malliongelman ratkaisuun ja Raviart-Thomas -elementtiä majorantin vapaan tensoriarvoisen funktion minimointiin. Elementtiratkaisijat ja virhearviot toteutettiin ohjelmointialustalla Matlab. Ohjelmakoodin tärkeimmät osat ovat tutkielman liitteessä A.

Majorantille \mathcal{M}_\oplus tehtyjen testien tulokset olivat odotusten mukaisia. Tapauksessa, jossa tarkka virhe tunnettiin, virheelle muodostui suurta yliarviointia. Tälle yliarvioinnille löydetään kuitenkin luonnollisia syitä. Tutkimusta voisi jatkaa etsimällä vakiolle κ_Ω tarkempia arvioita tutkittavissa alueissa sekä ratkaisemalla vapaata parametria τ tiheämmässä verkossa kuin malliongelma.

Uzawa-algoritmin toiminta osoittautui täysin odotusten mukaiseksi. Tässä tutkielmassa suoritettujen testien perusteella parametrin ρ arvo näyttäisi olevan suoraan verrannollinen konvergenssinopeuteen. Parametrin ei havaita vaikuttavan majorantin arvoon, vaan majorantit lähestyvät samaa arvoa parametrissa riippuvalla nopeudella.

Uutta tässä tutkimuksessa oli Repinin kehittämien funktionaalisten a posteriori virhearvioiden johtaminen ottaen huomioon Uzawa-algoritmin erityispiirteet sekä tulosten numeerinen testaaminen. Tämän tutkielman teoriaosuudesta on kirjoitettu artikkeli [1], jossa käsitellään lisäksi ns. augmentoitua Stokesin yhtälöä. Seuraavaksi tutkimusta laajennetaan Stokesin yhtälöstä Oseenin ongelmaksi, joka on yhden askeleen lähempänä Navier-Stokesin yhtälöä. Oseenin ongelmalle tullaan johtamaan vastaavia tuloksia ja toteuttamaan vastaavia numeerisia testejä.

Kirjallisuus

- [1] I. ANJAM, M. NOKKA, JA S. REPIN, *On a posteriori error bounds for approximations of the generalized Stokes problem generated by the Uzawa algorithm*, Russian J. Numer. Anal. Math. Modelling, 2012 (hyväksytty).
- [2] D. N. ARNOLD, F. BREZZI JA M. FORTIN, *A stable finite element for the Stokes equations*. Calcolo, 21(4):337-344 (1985), 1984.
- [3] D. BRAESS, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, 2nd edition, University Press, Cambridge, 2001 .
- [4] E.V. CHIZHONKOV JA M.A. OLSHANSKII, *On the domain geometry dependence of the LBB condition*, Mathematical Modellineng ja Numerical Analysis, Vol 34, No. 5, 2000, p 935-951.
- [5] P. G. CIARLET, *The finite element method for elliptic problems*, North Holland, New York, 1978.
- [6] M. DOBROWOLSKI, *On the LBB constant on stretched domains*, Mathematische Nachrichten, Vol 254-255, Issue 1, 2007, p 64-67.
- [7] M. FORTIN AND R. GLOWINSKI, *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*, North-Holland, New-York, 1983.
- [8] E. GORSHKOVA, *A posteriori error estimates and adaptive methods for incompressible viscous flow problem*, väitöskirja, Jyväskylän yliopisto, Jyväskylä, 2007.
- [9] E. R. HOPPE, *Finite element methods*, luentomoniste, http://www.math.uh.edu/~rohop/spring_11/downloads/Chapter7.pdf (2.5.2012).
- [10] C. O. HORGAN, L. E. PAYNE, *On inequalities of Korn, Friedrichs and Babuška-Aziz*, Arch. Ration. Mech. Anal. **82**(2), pp. 165–179, 1982.
- [11] T. KILPELÄINEN, *Mitta- ja integraaliteoria*, luentomoniste, <http://www.math.jyu.fi/opiskelu/monisteet/MATS110.pdf> (23.5.2012).
- [12] T. KILPELÄINEN, *Analyysi 1*, luentomoniste, <http://www.math.jyu.fi/opiskelu/monisteet/MATA111.pdf> (23.5.2012).

- [13] O. A. LADYZHENSKAYA, *The Mathematical Theory of Viscous Incompressible Flow*, Nauka, Moscow, 1970; English transl.:Gordon ja Breach, New York, 1969.
- [14] O. A. LADYZHENSKAYA JA V. A. SOLONNIKOV, *Some problems of vector analysis and generelized formulations of boundary value problems for the Navier-Stokes equations* [in Russian], Zap. Nauchn. Semin. LOMI 59 (1976), 81-116.
- [15] O. A. LADYZHENSKAYA JA N. N. URALTSEVA, *Linear and Quasilinear Elliptic Equations* [in Russian], Nauka, Moscow, 1973; English transl. of the 1st edition.: Academic Press, New York, 1968.
- [16] THE MATHWORKS, *Matlab*, <http://www.mathworks.se/products/matlab/> (30.4.2012).
- [17] THE MATHWORKS, *PDE Toolbox*, <http://www.mathworks.se/products/pde/> (30.4.2012).
- [18] S. G. MIKHLIN, *Constants in Some Inequalities of Analysis*, 1st edition, John Wiley & Sons Limited, 1986.
- [19] P. NEITTAANMÄKI JA S. REPIN, *Reliable methods for computer simulation. Error control and a posteriori error estimates*, Elsevier, Amsterdam, 2004.
- [20] G. PELOSI, *The finite-element method, Part I: R. L. Courant*, Antennas and Propagation Magazine, IEEE, 2007
- [21] P.A. RAVIART, J.M. THOMAS Primal hybrid finite element methods for 2nd order elliptic equations. Math. Comput. **31**, pp. 391–413 (1977)
- [22] S. REPIN, *A posteriori estimates for partial differential equations*, Walter de Gruyter, Berlin, 2008.
- [23] S. REPIN JA R. STENBERG, *A posteriori error estimates for the generalized Stokes problem*, Journal of Mathematical Sciences, Vol. 142, No. 1, 2007.
- [24] R. STENBERG, *Some problems in connection with the finite element solution of Navier-Stokes equations*, Proceedings of the Conference on Numerical Simulation Models, 1987.
- [25] R. TEMAM, *Navier-Stokes Equations*, 1st edition, North-Holland Publishing Company 1977.

[26] F. WHITE, *Fluid Mechanics*, 6th edition, McGraw-Hill, 2008.

A Lähdekoodit

Tähän tutkielmaan liittyvät ohjelmakoodit on tehty ohjelmointialustalla Matlab [16]. Tarvittavat elementtiratkaisijat on tehty itse, jotta voitiin kontrolloida ratkaisijoiden toimintaa paremmin. Matlabin lisäosaa PDE-toolbox [17] käytettiin elementtiverkkojen luomiseen ja käsittelyyn.

A.1 Malliongelman ratkaiseminen (solverMINI.m)

Tutkielman malliongelma ratkaistaan tiedostosta solverMINI.m löytyvällä ratkaisijalla. Nopeutta ratkaistaessa paine oletetaan tunnetuksi. Ratkaisijalle annetaan parametreina elementtiverkon ja integrointikvadratuurin tiedot, malliongelman parametrit, sekä paineen tietorakenne. Malliongelman ratkaisu palautuu yhtälön (6.1) mukaisen matriisiyhtälön ratkaisuun. Tästä eteenpäin "vasemmanpuoleisella matriisilla" viitataan yhtälön (6.1) matriisiin A , jota merkitään koodissa merkinnällä Mat ja vastaavasti "oikeanpuoleisella matriisilla" viitataan saman yhtälön matriisiin b , jota merkitään $Matf$.

```
% =====  
% Parametrit:  
% mesh:      elementtiverkon tietorakenne (t,p,e)  
% quadrature: tietorakenne, jossa tieto integrointikvadratuurista  
% pde:       ongelman tietorakenne (ODY:n kertoimet nyy, myy,  
%           lastifunktio f, mahdollisesti solverin (solverMINI)  
%           antama vasemmanpuoleinen matriisi Mat (mikäli ongelma  
%           ratkaistu ennenkin eri lastilla)  
% pressure:  paineen tietorakenne  
%  
% Ulos:  
% C:        ratkaisun kerroinvektori  
% Mat:      yhtälöryhmän vasemmanpuoleinen matriisi  
% Matf:     yhtälöryhmän oikeanpuoleinen vektori  
%  
% Hyödyntää funktiota:  
% quad.m  
% basis_functionsMINI.m  
% element_info.m  
% =====  
  
function [C, Mat, Matf] = solverMINI(mesh, quadrature, pde, pressure)
```

Integrointikvadratuuria hyödyntäen haetaan referenssielementissä olevien integrointipisteiden koordinaatit ja niihin liittyvät painot. Tämän jälkeen haetaan kantafunktioiden ja kantafunktioiden gradienttien arvot integrointipisteissä. Ennen matriisien kokoamisen aloittamista alustetaan yhtälön oikealle puolelle muodostuva matriisi nolaksi.

```

% Lasketaan kuvattavat int-pisteet X ja niihin liittyvät painot
% dim = käsiteltävä dimensio
% kertaluku = tarkkuus
[X,W] = quad(quadrature.deg, quadrature.dim);

% Haetaan kantafunktioiden arvot integrointipisteissä
[N, N_grad] = basis_functionsMINI(X);

% Alustetaan oikeapuoleinen matriisi.
koko = 2*(NOFnodes + NOFelems);
Matf = spdiags( zeros(koko,1) , 0, koko, 1);

```

Uzawan algoritmia hyödynnettessä vasemmanpuoleinen matriisi `Mat` tarvitsee koota vain ensimmäisellä iteraatiolla. Seuraavilla iteraatioilla ongelman parametreista vain paineessa on tapahtunut muutoksia, joten vain oikeanpuoleinen matriisi `Matf` tulee koota uudestaan. Mikäli vasemmanpuoleinen matriisi on koottu, on se tallennettu tietorakenteeseen `pde`. Ennen matriisien kokoamista testataan siis, että löytyykö kyseistä matriisia.

```

% Kun funktiota kutsutaan ensimmäisen kerran:
% Uzawan algoritmia käytettäessä yhtälöryhmän vasen puoli ei muutu
% lainkaan, joten on turhaa luoda vasemmanpuoleiset matriisit uudestaan
% jokaisella solverikutsulla. Tähän if-silmukkaan mennään VAIN jos
% vasemmanpuoleisia matriiseja ei ole luotu! Silmukka tekee sekä
% oikean- että vasemmanpuoleiset matriisit. Mikäli vasemmanpuoleiset
% matriisit oli jo luotu ennen funktiokutsua, tehdään uudelleen pelkkä
% yhtälöryhmän oikeanpuoleinen matriisi, sillä se muuttuu jokaisella
% kerralla.
%
% Tätä voi käyttää VAIN, jos verkkoa ei tihennetä eri iterointikertojen
% välissä. Jos tihennetään, tulee matriisin pde.Mat arvo asettaa aina
% nollaksi jokaisessa funktiokutsussa.
%
if pde.Mat == 0

```

Mikäli matriisia `Mat` ei löydy, alustetaan se nolllaksi. Aloitetaan käymään elementtejä läpi yksi kerrallaan. Alustetaan nolllaksi myös elementtikohtaiset lokaalit matriisit. Nimellä `Mat_lok_vasen1` kutsutaan lokaalia jäykkyysmatriisia, nimellä `Mat_lok_vasen2` lokaalia massamatriisia. Lokaalia oikeaan puoleen liittyvää matriisia nimitetään `Mat_lok_oikea`.

```

% Alustetaan globaali vasemmanpuoleinen matriisi
Mat = spdiags(zeros(koko,1) , 0, koko, koko);

% Silmukka käy läpi verkon kaikki elementit
for i =1:NOFelems
% Alustetaan lokaalit vasemmanpuoleiset matriisit

```

```

Mat_lok_vasen1 = zeros(size(N,1));
Mat_lok_vasen2 = zeros(size(N,1));

% Alustetaan lokaali oikeanpuoleinen matriisi
Mat_lok_oikea = zeros(size(N,1),1);

```

Seuraavaksi haetaan luvussa 6.4 esitetyn elementtikohtaiseen kuvaukseen F_K liittyvän matriisin B_K käänteismatriisi ja determinantti, sekä kuvauksella F_K kuvattujen integrointipisteiden koordinaatit käsiteltävässä elementissä.

```

% Haetaan tarpeellisia elementtikohtaisia tietoja funktiosta
% element_info.m, joka palauttaa int-pisteet kuvattuna työn alla
% olevaan elementtiin, kuvauksen komponentit matriisiin B_K ja vektorin
% b_K. Lisäksi palauttaa tarvittavat B_K:n käänteismatriisin ja
% determinantin.
[XY, ~, ~, B_K_inv, B_K_det] = element_info(mesh, i, quadrature);

```

Aloitetaan käymään läpi kantafunktioita kahden eri silmukan avulla.

```

% Silmukka käy läpi kaikki kantafunktiot
for j = 1:size(N,1)

% Silmukka käy läpi kaikki kantafunktiot
for k = 1:size(N,1)

```

Muodostetaan jäykkyysmatriisi. Koska aiemmin haettiin kantafunktioiden arvoja integrointipisteissä, tulee ne nyt kuvata käsiteltävään elementtiin, jolloin saadaan kantafunktioiden ja niiden gradienttien arvoja käsiteltävän elementin integrointipisteissä. Gradienttien tapauksessa kantafunktiot kerrotaan matriisiin B_K käänteismatriisin transpoosilla ja jaetaan sen determinantilla.

```

% Poistetaan "ylimääräinen" ykköstä oleva dimensio
gradJ = squeeze(N_grad(j,:,:,:));
gradK = squeeze(N_grad(k,:,:,:));

% Suoritetaan muuttujanvaihto ja tehdään numeerinen
% integrointi referenssielementissä.

% Lasketaan lokaaleja matriiseja integrointipiste
% kerrallaan (gradienttin kuvaaminen helpottuu)
for a=1:size(X,2)

    GradJ = gradJ(:,:,a)';

```

```

GradK = gradK(:,:,a)';

GradBJ = B_K_inv' * GradJ;
GradBK = B_K_inv' * GradK;

divJ = GradBJ(1,1) + GradBJ(2,2);
divK = GradBK(1,1) + GradBK(2,2);

%Jäykkyysmatriisi
Mat_lok_vasen1(j,k) = Mat_lok_vasen1(j,k) + ...
    abs(B_K_det) * pde.nu * ...
    W(a) * sum( sum( GradBJ .* GradBK ) ) ;

end

```

Muodostetaan massamatriisi. Kantafunktiot kuvataan vain jakamalla kuvauksen B_K determinantilla.

```

%Massamatriisi
W2 = zeros(2,length(W));
W2(1,:) = W;
W2(2,:) = W;

Mat_lok_vasen2(j,k) = abs(B_K_det) * ...
    sum(sum( ( pde.mu * (W2 .* squeeze(N(j,:,:)))' ) ...
    .* squeeze(N(k,:,:))' ) ) ;

end

```

Nyt vasempaan puoleen liittyvät lokaalit matriisit ovat valmiita, joten siirytään laskemaan oikeaan puoleen liittyvää lokaalia matriisia. Haetaan lastifunktion f ja paineen p arvoja käsiteltävän elementin integrointipisteissä. Tämän jälkeen voidaan laskea käsiteltävään elementtiin liittyvä lokaali oikeanpuoleinen matriisi.

```

% Muodostetaan elementtikohtainen lastivektori f parametrina tuodun
% inline-funktion avulla
f1 = pde.f1(XY(1,:),XY(2,:));
f2 = pde.f2(XY(1,:),XY(2,:));

f(1,:) = f1;
f(2,:) = f2;

% Lasketaan paineen arvoja integrointipisteissä
[p, ~] = solution_valuesMINI(mesh, pressure, i, XY, quadrature);

% Silmukka käy läpi integrointipisteet

```

```

summaf = 0;
summap = 0;

for n = 1:size(X,2)
    summaf = summaf + ( W(n) * sum(squeeze(N(j,:,n)) .* f(:,n)') );

    GradJ = gradJ(:, :, n)';
    GradBJ = B_K_inv' * GradJ;
    divJ = GradBJ(1,1) + GradBJ(2,2);

    summap = summap + (W(n) * divJ * p(n));
end

Mat_lok_oikea(j) = abs(B_K_det) * (summaf + summap);

end

```

Kun kaikki kantafunktiot on käyty läpi, voidaan sijoittaa lokaalit matriisit oikeille paikoille globaaliin matriisiin.

```

% Muodostetaan globaali matriisi sijoittamalla lokaalin matriisin
% arvot oikeille paikoille globaaliin matriisiin.
paikka = mesh.t(1:3,i);

paikka2 = [paikka(1), paikka(1) + NOFnodes + NOFelems, ...
    paikka(2), paikka(2) + NOFnodes + NOFelems, ...
    paikka(3), paikka(3) + NOFnodes + NOFelems, ...
    NOFnodes + i, 2*NOFnodes + NOFelems + i];

Mat(paikka2,paikka2) = Mat(paikka2,paikka2) + ...
    Mat_lok_vasen1 + Mat_lok_vasen2 + Mat_lok_vasen3;

Matf(paikka2) = Matf(paikka2) + Mat_lok_oikea;

```

Oletettiin, että nopeus reunalla on nolla, joten asetetaan nopeuden arvo nolaksi niissä solmuissa, jotka ovat alueen reunalla.

```

% Diricletin reunaehtojen lisääminen.
% Nopeus reunalla = 0
reuna = mesh.e(1,:);

for m = 1:size(reuna,2)

```

```

Mat(reuna(m), :) = 0;
Mat(reuna(m) + NOFnodes + NOFelems, :) = 0;

Mat(:, reuna(m)) = 0;
Mat(:, reuna(m) + NOFnodes + NOFelems) = 0;

Mat(reuna(m), reuna(m)) = 1;
Mat(reuna(m) + NOFnodes + NOFelems, reuna(m) ...
    + NOFnodes + NOFelems) = 1;

Matf(reuna(m)) = 0;
Matf(reuna(m) + NOFnodes + NOFelems) = 0;
end

```

Oletettiin, että nopeus reunalla on nolla, joten asetetaan nopeuden arvo nol- laksi niissä solmuissa, jotka ovat alueen reunalla. Nyt matriisiyhtälön ratkai- semiseen vaadittavat matriisit **Mat** ja **Matf** ovat valmiina.

Jos tämä ei ollut Uzawan algoritmin ensimmäinen iteraatio, riittää laskea oikeaan puoleen liittyvä matriisi **Matf**. Seuraavaksi esitettävä koodi on kopio juuri esitetystä, mutta siitä on poistettu kaikki oikeanpuolen kokoamiseen liittyvät operaatiot.

```

% Kun funktiota kutsutaan toisen tai useamman kerran
% Tähän silmukkaan mennään vain, jos vasemmanpuoleiset matriisit oli
% jo luotu ennen funktion kutsua. Tällöin on tarpeellista tehdä vain
% oikeanpuoleinen matriisi, joka siis muuttuu jokaisella ratkaisu-
% kerralla Uzawan algoritmia käytettäessä.

else
Mat = pde.Mat;

% Silmukka käy läpi verkon kaikki elementit
for i = 1:NOFelems

% Alustetaan lokaali oikeanpuoleinen matriisi
Mat_lok_oikea = zeros(size(N,1),1);

% Haetaan tarpeellisia elementtikohtaisia tietoja funktiosta
% element_info.m, joka palauttaa int-pisteet kuvattuna työn alla
% olevaan alkioon, kuvauksen komponentit matriisiin B_K ja vektorin
% b_K. Lisäksi palauttaa tarvittavat B_K:n käänteismatriisin ja
% determinantin.
[XY, ~, ~, B_K_inv, B_K_det] = element_info(mesh, i, quadrature);

% Silmukka käy läpi kaikki kantafunktiot
for j = 1:size(N,1)

```

```

% Muodostetaan elementtikohtainen lastivektori f parametrina tuodun
% inline-funktion avulla
f1 = pde.f1(XY(1,:),XY(2,:));
f2 = pde.f2(XY(1,:),XY(2,:));

f(1,:) = f1;
f(2,:) = f2;

% Lasketaan paineen arvoja integrointipisteissä
[p, ~] = solution_valuesMINI(mesh, pressure, i, XY, quadrature);

% Poistetaan "ylimääräinen" ykköstä oleva dimensio
gradJ = squeeze(N_grad(j,:,:,:));

% Silmukka käy läpi integrointipisteet
summaf = 0;
summap = 0;

for n = 1:size(X,2)
    summaf = summaf + ( W(n) * sum(squeeze(N(j,:,n)) .* f(:,n)') );

    GradJ = gradJ(:,:,n)';
    GradBJ = B_K_inv' * GradJ;
    divJ = GradBJ(1,1) + GradBJ(2,2);

    summap = summap + (W(n) * divJ * p(n));
end

Mat_lok_oikea(j) = abs(B_K_det) * (summaf + summap);

end

% Muodostetaan globaali matriisi sijoittamalla lokaalin matriisin
% arvot oikeille paikoille globaaliin matriisiin.
paikka = mesh.t(1:3,i);

paikka2 = [paikka(1), paikka(1) + NOFnodes + NOFelems, ...
    paikka(2), paikka(2) + NOFnodes + NOFelems, ...
    paikka(3), paikka(3) + NOFnodes + NOFelems, ...
    NOFnodes + i, 2*NOFnodes + NOFelems + i];

Matf(paikka2) = Matf(paikka2) + Mat_lok_oikea;

% Diricletin reunaehtoien lisääminen.
reuna = mesh.e(1,:);

for m = 1:size(reuna,2)
    Matf(reuna(m,:),:) = 0;
    Matf(reuna(m) + NOFnodes + NOFelems,:) = 0;
end

```



```
end
```

Seuraavaksi voidaan ratkaista yhtälön (6.1) mukainen yhtälöryhmä $\text{Mat} \cdot \text{C} = \text{Matf}$, joten malliongelma on ratkaistu.

```
% Ratkaistaan yhtälöryhmä Mat*C=Matf
C = Mat\Matf;
```

A.2 Tensoriarvoinen minimointiongelma (solverRT0t.m)

Luvun 5.3 mukaista minimointia tehdessä tarvitaan FEM-ratkaisija, joka tuottaa tensoriarvoisia ratkaisuja. Peruseriaatteeltaan tämä ratkaisija toimii samalla tavalla kuin edellä esitetty ratkaisija, mutta tässä vapausasteet ovat reunoja, saadut ratkaisut ovat tensoriarvoisia ja muuttujanvaihtoihin käytetään Piolamuunnoksia.

```
% =====
% Parametrit:
% mesh:      elementtiverkon tietorakenne (t,p,e)
% quadrature: tietorakenne, jossa tieto integrointikvadratuurista
% pde:      ongelman tietorakenne (ODY:n kertoimet nyy, myy,
%           lastifunktio f
% velocity: nopeuden tietorakenne
% pressure: paineen tietorakenne
%
% Ulos:
% L1: jäykkyyismatriisi
% L2: massamatriisi
% R1: ensimmäinen oikeanpuolen matriisi
% R2: toinen oikeanpuolen matriisi
%
% Ongelman ratkaisu palautuu yhtälöryhmän
% (L1 + L2)*C = (R1 + R2)
% ratkaisuun
%
% Hyödyntää funktiota:
% quad.m
% basis_functionsRT0t.m
% element_info.m
% =====

function [L1, L2, R1, R2] = solverRT0t(mesh, quadrature, pde,...
                                       velocity, pressure)

% Lasketaan kuvattavat int-pisteet X ja niihin liittyvät painot
```

```

% dim = käsiteltävä dimensio
% deg = tarkkuus
[X,W] = quad(quadrature.deg, quadrature.dim);

% Haetaan kantafunktioiden arvot integrointipisteissä
[N, N_div] = basis_functionsRTO(X, 't');

% Haetaan "reunatiedot"
[element2edges, edge2nodes,~] = edgedata(mesh.e,mesh.t,2);
NOFedges = size(edge2nodes,2);
NOFelems = size(element2edges,2);

% Alustetaan globaali oikeanpuoleinen matriisi
% Matriisien rivimäärät = vapausasteiden lukumäärä.
koko = 2*NOFedges;
R1 = spdiags( zeros(koko,1) , 0, koko, 1);
R2 = spdiags( zeros(koko,1) , 0, koko, 1);

% Alustetaan globaali vasemmanpuoleinen matriisi
L1 = spdiags(zeros(koko,1) , 0, koko, koko);
L2 = spdiags(zeros(koko,1) , 0, koko, koko);

% Silmukka käy läpi verkon kaikki elementit
for i =1:NOFelems

% Lasketaan merkit
tmp = mesh.t([1 2 3],i) - mesh.t([2 3 1],i);
elemsign = tmp ./ abs(tmp);
elemsigns(1) = elemsign(1);
elemsigns(2) = elemsign(1);
elemsigns(3) = elemsign(2);
elemsigns(4) = elemsign(2);
elemsigns(5) = elemsign(3);
elemsigns(6) = elemsign(3);

% Alustetaan lokaalit vasemmanpuoleiset matriisit
L1_lok = zeros(size(N,1));
L2_lok = zeros(size(N,1));

% Alustetaan lokaali oikeanpuoleinen matriisi
R1_lok = zeros(size(N,1),1);
R2_lok = zeros(size(N,1),1);

% Haetaan tarpeellisia elementtikohtaisia tietoja funktiosta
% element_info.m, joka palauttaa int-pisteet kuvattuna työn alla
% olevaan elementtiin, kuvauksen matriisin B_K ja sen determinantin.
[XY, B_K, ~, ~, B_K_det] = element_info(mesh, i, quadrature);

% Silmukka käy läpi kaikki kantafunktiot
for j = 1:size(N,1)

    for k = 1:size(N,1)

% Suoritetaan muuttujanvaihto ja tehdään numeerinen

```

```

% integrointi referenssielementissä.

% Div elementteittäin vakio => ei tarvita painoja => ref-
% elementin pinta-alalla kertominen riittää
    divJ = N_div(j,:);
    divK = N_div(k,:);

% Jäykkyydsmatriisi
    L1_lok(j,k) = 0.5/abs(B_K_det) * pde.nu ...
        * sum((elemsigns(j)* divJ) .* (elemsigns(k) * divK));

% Massamatriisi
    for piste = 1:size(X,2)
        L2_lok(j,k) = L2_lok(j,k) + 1/pde.nu * ...
            (1/abs(B_K_det)) * sum(sum( ( pde.mu * W(piste) * ...
                (elemsigns(j) * (B_K * squeeze(N(j,:,: ,piste)))) ...
                .* (elemsigns(k) * (B_K * squeeze(N(k,:,: ,piste)))) )' )' )' );
    end

    end

% Oikeanpuolen laskeminen:

% Muodostetaan elementtikohtainen lastivektori f parametrina tuodun
% inline-funktion avulla
    f1 = pde.f1(XY(1,:),XY(2,:));
    f2 = pde.f2(XY(1,:),XY(2,:));
    f(1,:) = f1;
    f(2,:) = f2;

    divJ = N_div(j,:);
    for n = 1:size(X,2)
        R1_lok(j) = R1_lok(j) + (elemsigns(j))*...
            (W(n) * divJ * (f(:,n)
                - squeeze(pde.mu * velocity.values(i,n,:))));
    end

    q(:,1,1) = pressure.values(i,:);
    q(:,2,2) = pressure.values(i,:);

    end

    for piste = 1:size(X,2)
        R2_lok(j) = R2_lok(j) + 1/pde.nu * ...
            sum(sum( elemsigns(j) * W(piste) ...
                *((B_K * squeeze(N(j,:,: ,piste)))) ...
                .* (squeeze( -pde.nu*velocity.grad_values(i,piste,:,:)) ...
                    + squeeze(q(piste,:,:)) )' )' )' );
    end

```

```

end

% Muodostetaan globaali matriisi sijoittamalla lokaalin matriisin
% arvot oikeille paikoille globaaliin matriisiin.
paikka = element2edges(1:3,i);
paikka2 = [paikka(1), paikka(1) + NOFedges, ...
           paikka(2), paikka(2) + NOFedges, ...
           paikka(3), paikka(3) + NOFedges];

L1(paikka2,paikka2) = L1(paikka2,paikka2) + L1_lok;
L2(paikka2,paikka2) = L2(paikka2,paikka2) + L2_lok;

R1(paikka2) = R1(paikka2) + R1_lok;
R2(paikka2) = R2(paikka2) + R2_lok;

end

end

```

A.3 Majorantin laskeminen

Lasketaan arvoja majorantille, joka on esitetty lauseessa 5.2.2.

Aluksi määritellään tutkittava alue ja muodostetaan verkko. Molemmissa tutkittavissa alueissa (yksikköneliö ja yksikköympyrä) käytetään epäsäännöllistä verkkoa.

```

function [] = Majorantti_Uzawa()

[p,e,t] = initmesh('unitsquareg', 'Hmax', 0.055);

mesh.p = p;
mesh.e = e;
mesh.t = t;

[~, edge2nodes, ~] = edgedata(mesh.e,mesh.t,2);
NOFedges = size(edge2nodes,2);

```

Mikäli tarkat ratkaisut tiedetään, sijoitetaan ne tietorakenteeseen `exact` inline-funktioina. Muuten tehdään `exact`:sta muuttuja ja annetaan sen arvoksi nolla.

```

% Tarkat ratkaisut tiedetään:

```

```

exact.u1 = inline('20*x.^2.*y.*(2*y - 1).*(x - 1).^2.*(y - 1)');
exact.u1_gradx = inline('40*x.*y.*(2*y - 1).*(x - 1).^2.*(y - 1) ...
+ 20*x.^2.*y.*(2*x - 2).*(2*y - 1).*(y - 1)');
exact.u1_grady = inline('20*x.^2.*y.*(2*y - 1).*(x - 1).^2 ...
+ 20*x.^2.*(2.*y - 1).*(x - 1).^2.*(y - 1) ...
+ 40*x.^2.*y.*(x - 1).^2.*(y - 1)');
exact.u2 = inline('-20*x.*y.^2.*(2*x - 1).*(x - 1).*(y - 1).^2');
exact.u2_gradx = inline('- 20*x.*y.^2.*(2*x - 1).*(y - 1).^2 ...
- 20*y.^2.*(2*x - 1).*(x - 1).*(y - 1).^2 ...
- 40*x.*y.^2.*(x - 1).*(y - 1).^2');
exact.u2_grady = inline('- 40*x.*y.*(2*x - 1).*(x - 1).*(y - 1).^2 ...
- 20*x.*y.^2.*(2*x - 1).*(2*y - 2).*(x - 1)');
exact.p = inline('2*x - 1');
% % tarkkaa ratkaisua ei tiedetä
% exact = 0;

```

Määritellään ongelman ja integrointikvadratuurin parametrit, elementtien tyypit (paineelle epäjatkuva kolmioelementti ja nopeudelle MINI-elementti), paineen alkuarvo, Uzawan parametri ρ , Friedrichin vakio ja LBB-vakion käänteisluku.

```

pde.nu = 1;
pde.mu = 1;
pde.Mat = 0;

pde.f1 = inline('x.*y.*(x - 1).*(y - 1) - 2*y.*(y - 1) ...
- 2*x.*(x - 1) + 2');
pde.f2 = inline('-(x - y).*(- x.^2.*y.^2 + x.^2.*y + 2*x.^2 ...
+ x.*y.^2 - 5*x.*y + 2*y.^2 - 2)');

quadrature.dim = 2;
quadrature.deg = 8;

[X,W] = quad(quadrature.deg, quadrature.dim);

pressure.elemtype = 'tri_discont';
pressure.dof = zeros(size(mesh.t,2),3);

velocity.elemtype = 'mini';

% Uzawan parametri
uzawa.rho = 2*pde.nu - 0.1;

```

```

% Käytetään Fiedrichin vakiona 1/(pi*sqrt(2))
F = 1/(sqrt(2)*pi);

% Käytetään LBB-vakion käänteislukuna
kappa = 2.6131;

constant = kappa*sqrt(F^2*pde.mu + pde.nu);

```

Aloitetaan Uzawa-iteraatioiden teko. Iteraatioita tehdään maksimissaan itermax kpl.

```

disp(' ')
disp(' ')
disp(' ')
disp('#####')
disp('# Verkko: ')
disp(['# Elementtejä ', num2str(size(mesh.t,2)), ' kpl, solmuja ', ...
      num2str(size(mesh.p,2)), ' kpl, reunoja ', num2str(NOFedges), ' kpl'])
disp('# ')
disp('# ODY:n parametrin:')
disp(['# Nyy: ', num2str(pde.nu), ', Myy: ', num2str(pde.mu)])
disp('# ')
disp('# Uzawan parametri: ')
disp(['# Rho ', num2str(uzawa.rho),'])
disp('#####')
disp(' ')

iter = 0;
itermax = 20;
Majorant = zeros(itermax,1);
exact_error = zeros(itermax,1);

while (iter < itermax)
    iter = iter + 1;

```

Jokaisella iteraatiolla tehdään yksi Uzawa-askel, joka jälkeen on saatu likiarvo nopeudelle ja päivitetty likiarvo paineelle. Lasketaan nopeuden ja nopeuden gradientin sekä paineen arvoja integrointipisteissä.

```

[Mat, velocity, pressure] = UzawaMINI(mesh, quadrature, pde, ...
                                     uzawa, velocity, pressure);
pde.Mat = Mat;

val = zeros(size(mesh.t,2), size(X,2), 2);

```

```

grad = zeros(size(mesh.t,2),size(X,2),2,2);
p_val = zeros(size(mesh.t,2),size(X,2));

for i = 1:size(t,2)
    [values, gradvalues] = solution_valuesMINI(mesh, velocity, i,...
                                                X, quadrature);
    val(i,:,:)= values;
    grad(i,:,:)= gradvalues;

    [p_values,~] = solution_valuesMINI(mesh, pressure, i, X, ...
                                        quadrature);
    p_val(i,:) = p_values;
end

velocity.values = val;
velocity.grad_values = grad;
velocity.div_values = grad(:,:,1,1) + grad(:,:,2,2);
pressure.values = p_val;

```

Mikäli tarkka ratkaisu tiedetään, lasketaan likiarvon tarkka virhe energianormissa, joka on esitetty yhtälössä (3.4).

```

if isstruct(exact)
    [Uerr] = UzawaVirhe(mesh, velocity, quadrature, exact, pde);
    exact_error(iter) = Uerr;
    disp(['* Uzawan iteraatio ', num2str(iter), ...
         ' * Tarkka virhe: ', num2str(Uerr)])
else
    disp(['* Uzawan iteraatio ', num2str(iter), ' *'])
end

```

Seuraavaksi minimoidaan τ β :n arvolla 1.

```

% Oletetaan, että \beta = 1, ja ratkaistaan matriisit
[L1,L2,R1,R2] = solverRT0t(mesh, quadrature, pde, velocity, ...
                            pressure);

% \beta:lle alkuarvo ennen iteraatioiden aloittamista
beta = 1;

```

Minimoidaan β :n ja τ :n arvoja vuorotellen kappaleessa 5.3 esitetyn rutiinin mukaisesti ja lasketaan majorantille (lause 5.2.2) arvo jokaisen minimoinnin jälkeen.

```

for k=1:5

    H = (F^2*(1+beta))/(F^2*pde.mu*(1+beta) + pde.nu);

    tau.dof = (H*L1 + (1 + 1/beta)*L2)\(- H*R1 - (1 + 1/beta)*R2);
    tau.values = zeros(size(mesh.t,2),size(X,2),2,2);
    tau.div_values = zeros(size(mesh.t,2),size(X,2),2);
    for element = 1:size(mesh.t,2)
        [values, div_values] = solution_valuesRTO(mesh, tau, ...
            element, X, quadrature,'t');
        tau.values(element,:,:,:) = values;
        tau.div_values(element,:,:,:) = div_values;
    end

    div_norm_elem = zeros(size(mesh.t,2),1);

    M1 = zeros(size(mesh.t,2),1);
    M2 = zeros(size(mesh.t,2),1);

    for i=1:size(mesh.t,2)
        [XY, ~, ~, ~, B_K_det] = element_info(mesh, i, quadrature);

        f1 = pde.f1(XY(1,:),XY(2,:));
        f2 = pde.f2(XY(1,:),XY(2,:));

        f(:,1) = f1;
        f(:,2) = f2;

        q(:,1,1) = pressure.values(i,:);
        q(:,2,2) = pressure.values(i,:);

        M1(i) = abs(B_K_det) * sum(W .* sum(((f + squeeze(-pde.nu...
            * velocity.values(i,:,:)) ...
            + tau.div_values(i,:,:)).^2),2));

        M2(i) = abs(B_K_det) * sum(W .* sum(sum(((1/sqrt(pde.nu)...
            * (squeeze(tau.values(i,:,:,:) - pde.nu ...
            * velocity.grad_values(i,:,:,:))+ q)).^2),3),2));

        div_norm_elem(i) = abs(B_K_det) ...
            * sum(W'.*(velocity.div_values(i,:).^2));
    end

    % Duaalin normin neliö
    D = sum(M2);

    % Residuaalin normin neliö
    R = sum(M1);

    % Tähän lasketaan u^k-u^k_h
    Maj = H*R + (1+1/beta)*D ;

```



```

div_norm = sqrt(sum(div_norm_elem));

% Koko majorantti
Majorantti = 2*constant*div_norm ...
            + (2*constant/sqrt(pde.nu) + 1)*sqrt(Maj);

disp(['Iteraatio: ', num2str(k), ...
      ' | Betan arvo: ', num2str(beta) ,...
      ' | Majorantti: ', num2str(Majorantti), ...
      ' | Div: ', num2str(div_norm),...
      ' | R^2: ', num2str(R), ' | D^2: ', num2str(D) ])

beta1 = -(pde.mu*D)*(pde.mu*F^2/pde.nu + 1) - sqrt(D*R) ...
        *(F*pde.mu/sqrt(pde.nu) + sqrt(pde.nu)/F);
beta2 = (F^2*D)*(pde.mu/pde.nu) - R;
beta = beta1/beta2;

```

Iteraatiot lopetetaan, mikäli β :n arvo kasvaa suhteettoman suureksi.

```

if beta > 1000000
    disp('betan arvo yli 1 000 000 => lopetetaan iteraatiot')
    break
end
end

```

Kun minimoinnit on saatu valmiiksi, saadaan tämän Uzawan askeleen lopullinen majorantin arvo.

```

Majorant(iter) = Majorantti;

disp(' ')
end

end

```