

Joel Pohjolainen

KONTROLLITILAVUUSMENETELMÄN ADAPTIIVISUUS, VIRHELÄHTEET JA VIRHEARVIOT

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Toukokuu 2019

TIIVISTELMÄ

Joel Pohjolainen: Kontrollitulavuusmenetelmän adaptiivisuus, virhelähteet ja virhearviot
Diplomityö
Tampereen yliopisto
Konetekniikan diplomi-insinöörin tutkinto-ohjelma
Toukokuu 2019

Diplomityössä tutkittiin sitä, että miksi virtauslaskennan simulaatiotuloksissa esiintyy diskretisointivirhettä, miten sen suuruutta voidaan arvioida ja kuinka sitä voidaan mahdollisimman tehokkaasti vähentää. Diskretisointivirheiden kertaluokasta saadaan arvioita tutkimalla laskennassa käytettyjä diskretisointimenetelmiä, näiden stabiiliutta ja myös sitä, miten ne reagoivat laskentaverkkojen topologiaan virheisiin. Virtauslaskennassa käytetyssä kontrollitulavuusmenetelmässä pyritään yleensä siihen, että diskretisointivirhe pienenee neliöllisesti laskentaverkkoa tihennettäessä, mutta todellisen diskretisointivirheen tason arvioimiseen tarvitaan kuitenkin jälkikäteisvirhearvio. Parhaimmaksi tavaksi diskretisointivirheen pienentämiseen löydettiin jälkikäteisvirhearviota hyödyntävä h-adaptiivinen algoritmi, jolla voidaan lisätä laskentaverkon resoluutiota sopivissa paikoissa ja näin ollen tulosten virhetaso saadaan jyrkästi putoamaan. Toisena vaihtoehtona työssä pohdittiin tulosten virhetason pienentämistä lisäämällä runsaasti laskentatehoa, mikä mahdollistaa tiheämpien laskentaverkkojen käyttämisen, mutta tätä vaihtoehtoa ei pidetty hyvänä pitkän aikavälin ratkaisuna.

Kontrollitulavuusmenetelmän diskretisointivirheestä saadaan kohtuullinen kuva käsittelemällä ratkaistavissa yhtälöissä esiintyvien termien diskretisointia yksi termi kerrallaan. Diffuusiotermien diskretisointi keskeisdifferenssimenetelmällä toisen kertaluvun tarkkuudella on kohtuullisen suoraviivaista, kun laskentaverkon epäortogonaalisuus otetaan huomioon. Sen sijaan toisen kertaluvun konvektiomenetelmien ongelmana on niiden epästabiilius. NVD-viitekehukseen pohjautuva Gamma-menetelmä havaittiin parhaaksi konvektiomenetelmäksi, sillä kyseinen menetelmä tuottaa tuloksia pääasiassa toisen kertaluvun tarkkuudella ja lisää vain minimimäärän numeerista diffuusiota ratkaisuun. Gradienttitermien diskretisoinnissa yleensä jäädään halutusta toisen kertaluvun tarkkuudesta, ellei laskentaverkko ole säännöllinen tai siinä esiintyy vinoumaa. Painotetun pienimmän neliösumman gradienttimenetelmä yleensä kärsii näistä vähiten ja menetelmä yleensä tuottaa gradientteja ensimmäisen kertaluvun tarkkuudella myös epäsäännöllisissä laskentaverkoissa. Lähdetermien osalta kontrollitulavuusmenetelmä on luonnostaan tarkkuudeltaan toista kertalukua. Tavoiteltu toisen kertaluvun tarkkuus ei menetelmien yhteisvaikutuksesta johdu aina täyty.

Virtauslaskennan numeeristen ratkaisujen virhetasoa tutkittiin erilaisilla kontrollitulavuusmenetelmään johdetuilla jälkikäteisvirhearvioilla, kuten momenttivilhe- ja jäännösvirhemenetelmällä. Virhearvioiden havaittiin antavan hieman toisistaan poikkeavia tuloksia niin käytännössä kuin teoriassa. Realistisissa käyttökohteissa erityisesti jäännösvirhemenetelmän tunnetaan arvioivan diskretisointivirheen suuruutta hyvin. Adaptiivisen algoritmin suorituskyky ei kuitenkaan merkittävästi riippunut valitusta virhearviomenetelmästä ja adaptiivinen algoritmi osoittautui erittäin hyväksi tavaksi vähentää diskretisointivirhettä insinöörisovelluksessa. Diskretisointivirheen tilavuuskeskiarvon pienentämisessä adaptiivisuus on jopa kertaluokkaa parempi kuin perinteinen laskentaverkon mallinnusmetodi, jossa mallinnetaan rajakerrokset tarkasti. Algoritmin toteuttamiseen valittu OpenFOAM v5.0 -virtauslaskentasoftware osoittautui hyväksi kehitysalustaksi adaptiiviselle algoritmille ja algoritmi saatiin toteutettua kohtuullisessa ajassa.

Avainsanat: adaptiivisuus, diskretisointivirhe, kontrollitulavuusmenetelmä, OpenFOAM, PIMPLE-algoritmi, virtauslaskenta, virhearvio, virhelähde

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Joel Pohjolainen: Adaptivity, error sources and error estimates in finite volume method
Master of Science Thesis
Tampere University
Master's Degree Programme in Mechanical Engineering
May 2019

This Master's thesis investigates why simulation results in computational fluid dynamics are affected by discretisation errors, how these errors can be estimated and most efficiently reduced. The order of magnitude of these discretisation errors can be found by researching the different discretisation methods that are used during simulation and examining their stability and how they respond to topological errors. Well established practice of computational fluid dynamics indicates that these discretisation errors usually diminish quadratically as the mesh size is reduced but a more accurate evaluation of discretisation errors require the use of an a posteriori error estimate. The best way to reduce such errors is to use an h-adaptive algorithm that is driven by appropriate error estimates, which permits rapid reduction of the discretisation error as mesh resolution is increased only where it is needed. Another way to reduce discretisation errors is to significantly increase the available computational power, which allows for much finer mesh size to be used. However, this idea was quickly abandoned for its unsustainability in long term use.

A preliminary a priori estimate of the discretisation error present in the simulation results can be found when error sources of different discretisation methods are evaluated method by method. Discretisation of diffusion terms is generally straight forward with central difference method and 2nd order accuracy can be easily shown if nonorthogonality is handled appropriately but 2nd order convection discretisation methods on the other hand are unstable by nature. To resolve this, Gamma-method within the NVD-framework was found to be the most attractive method as this method produces nearly 2nd order results with limited numerical diffusion. Different gradient methods show remarkable difficulty in reaching 2nd order accuracy unless the used mesh is structured or does not have any significant skewness. Usually, the weighted least squares method suffers least from these and produces 1st order accurate gradients even in unstructured cases. To the contrary, evaluation of source terms in 2nd order accuracy is straightforward but due to the overall discrepancy of the different discretisation methods, simulation results in computational fluid dynamics might not always reach the desired 2nd order accuracy.

In this study, a more accurate estimation of the discretisation error of the numerical solutions was done by using a posteriori error estimates that fall within the framework of the finite volume method such as the moment error estimate and the residual error estimate. Both error estimates produce distinct results and the difference between these error estimates is known in literature as well. In realistic situations, the residual error estimate can be trusted to estimate the discretisation error with reasonable accuracy. An adaptive process driven by either error estimate proved out to be an excellent method to reduce discretisation errors in a realistic case study. Up to an order of magnitude benefit can be found over the traditional mesh modelling method where user manually puts more resolution in wanted places such as boundary layers. Chosen OpenFOAM v5.0 platform turned out to be a very capable option for the practical application of the studied h-adaptive algorithm and the algorithm was thus implemented in a reasonable time schedule.

Keywords: adaptivity, discretisation error, finite volume method, OpenFOAM, PIMPLE-algorithm, computational fluid dynamics, error estimate, error source

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä diplomityö tehtiin sen vuoksi, että virtauslaskennan insinöörisovelluksissa simulaatiotulosten todellista virhetasoa on vaikea arvioida tarkasti ja työn avulla haluttiin päästä eroon erilaisista kokemusperäisistä virhearviomitoista. Samalla virtauslaskennan taitoni kehittyivät ja sain kokemusta OpenFOAM-virtauslaskentasovelluksen ohjelmoinnista. Työ päädyttiin kirjoittamaan suomen kielellä siksi, että aiheesta saadaan tietoa myös suomeksi. Teoriaosuus osoittautui varsinaiseksi seikkailuksi mekaniikan maailmaan, ja tietoa kertyi huomattavan paljon.

Haluaisin kiittää työni tarkastajia professori Reijo Kouhiaa ja apulaisprofessori Jarkko Niirasta, jotka ohjasivat työni etenemistä ja antoivat työni sisältöön liittyviä kommentteja. Kiitän myös Medikro Oy:tä, jolta sain työssä käytetyn 3D-mallin sekä kokeellisen datan käyttööni. Suuri kiitos kuuluu myös perheelleni, joka tuki suuresti diplomityöni edistymistä.

“Equipped with his five senses, man explores the universe around him and calls the adventure Science.” (Edwin Powell Hubble)

Tampereella, 29.5.2019

Joel Pohjolainen

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Työn tausta, tutkimuskysymykset ja tutkimusmenetelmät	1
1.2	OpenFOAM-virtauslaskentasovellus	3
2.	DIFFERENTIAALIYHTÄLÖIDEN DISKRETISOINTI KONTROLLITILAVUUSMENETELMÄLLÄ	5
2.1	Kontrollitilavuusmenetelmän periaate	5
2.2	Ratkaistavat yhtälöt	8
2.3	Diffuusiotermin	13
2.4	Gradienttitermi	17
2.5	Konvektiotermi.....	25
2.6	Aikatermi.....	36
2.7	Lähdetermi	37
2.8	SIMPLE- ja PISO-algoritmit sekä muut ratkaisumenetelmät	38
3.	NAVIERIN–STOKESIN YHTÄLÖIDEN VIRHEARVIOMENETELMÄT.....	43
3.1	Virhearviomenetelmien perusta	43
3.2	Diskretisointivirheen arvioiminen Taylorin sarjakehitelmän avulla	51
3.3	Momenttimenetelmä.....	53
3.4	Jäännös menetelmä.....	55
3.5	Tasapainotettujen jäännösten menetelmä.....	58
4.	ADAPTIIVISEN KONTROLLITILAVUUSMENETELMÄN OPENFOAM- TOTEUTUS	66
4.1	OpenFOAM-alustan valmiiden ominaisuuksien selvitys.....	66
4.2	Lisäominaisuuksien ohjelmointi.....	69
4.3	Laskenta-asetusten valinta.....	73
5.	TOTEUTETUN MENETELMÄN KÄYTÄNNÖNSOVELLUS	77
5.1	Esimerkkitapaus ja ongelman asettelu	77
5.2	Menetelmän tulokset ja vertailu perinteisiin metodeihin	79
5.3	Kokeellisen datan ja virhearvioiden vertailu.....	86
6.	YHTEENVETO	89
	LÄHTEET.....	93

KUVALUETTELO

Kuva 1.1	<i>Virtauslaskennan (OpenFOAM) skaalaavuus kaksi Intel XEON-E5 2687W -prosessoria sisältävillä työasemilla (Keough 2014, s. 19).....</i>	<i>1</i>
Kuva 1.2	<i>Virtauslaskennassa käytetty h-adaptiivinen laskentaverkko (Muzafferija & Gosman 1997, s. 782).....</i>	<i>3</i>
Kuva 1.3	<i>OpenFOAM-virtauslaskentasovelluksen rakenne (Greenshields 2017, s. 17).....</i>	<i>4</i>
Kuva 2.1	<i>Tyypillinen laskentaverkko, yksittäisen kontrollitilavuuden säilymisominaisuus sekä kontrollitilavuutta ympäröivät naapurit (Moukalled et al. 2016, s. 109).....</i>	<i>5</i>
Kuva 2.2	<i>Kahden kontrollitilavuuden välisen pinnan epäortogonaalisuus ja tätä muotovirhettä kompensoiva ylirelaksoitu korjausmenetelmä (Moukalled et al. 2016, s. 243).....</i>	<i>13</i>
Kuva 2.3	<i>Kontrollitilavuuksien välisen pinnan vinouma (Moukalled et al. 2016, s. 254).....</i>	<i>18</i>
Kuva 2.4	<i>Säännöllisistä nelikulmioista kolmioitu laskentaverkko, jossa Greenin–Gaussin gradientti menettää tarkkuuttaan. Mukailtu lähteestä (Syrakos et al. 2017, s. 8).....</i>	<i>20</i>
Kuva 2.5	<i>Gradienttimenetelmien tutkimisessa käytettyjä laskentaverkkoja (Syrakos et al. 2017, s. 26).....</i>	<i>23</i>
Kuva 2.6	<i>Pienimmän neliösumman menetelmän ja Greenin–Gaussin menetelmän keskimääräisen virheen ja maksimivirheen käyttäytyminen (Syrakos et al. 2017, s. 27).....</i>	<i>23</i>
Kuva 2.7	<i>Rajapintakontrollitilavuuksien aiheuttama haaste pienimmän neliösumman menetelmälle. Mukailtu lähteestä (Syrakos et al. 2017, s. 23).....</i>	<i>24</i>
Kuva 2.8	<i>Mielivaltaisten kontrollitilavuuksien välinen rajapinta f (Moukalled et al. 2016, s. 457).....</i>	<i>25</i>
Kuva 2.9	<i>1D-laskentaverkko ja konvektiotermin lineaarinen variaatio kontrollitilavuuksien välillä (Moukalled et al. 2016, s. 370).....</i>	<i>26</i>
Kuva 2.10	<i>Keskeisdifferenssimenetelmän, ylävirran menetelmän ja analyyttisen ratkaisun vertailu (Moukalled et al. 2016, s. 378).....</i>	<i>27</i>
Kuva 2.11	<i>Ylävirran menetelmä (Moukalled et al. 2016, s. 376).....</i>	<i>28</i>
Kuva 2.12	<i>Lineaarinen ylävirran menetelmä (Moukalled et al. 2016, s. 390).....</i>	<i>29</i>
Kuva 2.13	<i>Vuorajoittimien TVD-alue. Mukailtu lähteestä Sweby (1984, s. 1001).....</i>	<i>30</i>
Kuva 2.14	<i>Swebyn ja Van Leerin ehdottamat rajoitinfunktiot. Mukailtu lähteestä (Sweby 1984, s. 1002).....</i>	<i>31</i>
Kuva 2.15	<i>Vasemmalla CBC-rajoitusehto NVD-kuvaajassa ja oikealla yleisimmät konvektiomenetelmät NVD-viitekehyksessä. Mukailtu lähteestä (Jasak 1996, s. 102).....</i>	<i>33</i>

Kuva 2.16	<i>Gamma-menetelmä NVD-viitekehyksessä. Mukailtu lähteestä (Jasak 1996, s. 110).</i>	35
Kuva 3.1	<i>Parannetun gradientin menetelmä. Mukailtu lähteestä (Ainsworth & Oden 1997, s. 17).</i>	46
Kuva 3.2	<i>Tyypillinen elementtimenetelmän ja kontrollitulavuusmenetelmän duaalimenetelmässä käytetty laskentaverkko. Mukailtu lähteestä (Ferziger & Perić 2002, s. 245).</i>	50
Kuva 3.3	<i>Kontrollitulavuuksien välillä esiintyvä jäännös. Mukailtu lähteestä (Jasak 1996, s. 174).</i>	56
Kuva 4.1	<i>Adaptiivinen PIMPLE-algoritmi</i>	67
Kuva 5.1	<i>Testigeometrian a) sisäänmeno, b) ulostulo ja c) symmetriataso</i>	77
Kuva 5.2	<i>Kahden eri virhe-estimaatin avulla luodut adaptiiviset laskentaverkot.</i>	80
Kuva 5.3	<i>Analysoitujen laskentaverkkojen virhe-estimaatit</i>	81
Kuva 5.4	<i>Nopeuskentän muuttuminen adaptiivisuuden edetessä</i>	83
Kuva 5.5	<i>Painekentän paraneminen adaptiivisuuden vaikutuksesta</i>	84
Kuva 5.6	<i>Tulosten diskretisointivirheiden tilavuuskeskiarvot ja maksimi-arvot</i>	85
Kuva 5.7	<i>Ratkaistujen laskentaverkkojen epäortogonaalisuus ja vinouma</i>	86
Kuva 5.8	<i>Painegradientin suhteellinen virhe</i>	87
Kuva 5.9	<i>Rakovirtauksen turbulenssin kasvaminen eri menetelmillä</i>	88

LYHENTEET JA MERKINNÄT

CBC	engl. Convection Boundedness Criterion, konvektion rajoitusehto
CD	engl. Central Difference scheme, keskeisidifferenssimenetelmä
DWF-menetelmä	engl. Downwind Weighing Factor Method, alavirran painoker- toimien menetelmä
FOAM	engl. Field Operation and Manipulation, virtauslaskentasovellus
GCI	engl. Grid Convergence Index, laskentaverkon suppenemisindeksi
HPC	engl. High-Performance Computing, suurteholaskenta
LES	engl. Large Eddy Simulation, suurten pyörteiden menetelmä
LUD	engl. Linear Upwind scheme, lineaarinen ylävirran menetelmä
NVA	engl. Normalised Variable Approach, normalisoidun muuttujan me- netelmä
NVD	engl. Normalised Variable Diagram, normalisoitujen muuttujien kaavio
OpenFOAM	engl. Open Source Field Operation and Manipulation, avoimen läh- dekoodin virtauslaskentasovellus
PIMPLE	engl. Merged PISO-SIMPLE, yhdistetty PISO- ja SIMPLE-algoritmi
PISO	engl. Pressure-Implicit with Splitting of Operators, implisiittinen paineenratkaisualgoritmi
QUICK	engl. Quadratic Upstream Interpolation for Convective Kinematics, neliöllinen ylävirran menetelmä
RANS-yhtälöt	engl. Reynolds Averaged Navier–Stokes equations, Reynoldsin keskiarvoistetut Navierin–Stokesin yhtälöt
SIMPLE	engl. Semi-Implicit Method for Pressure-Linked Equations, puoli- implisiittinen paineenratkaisualgoritmi
SIMPLEC	engl. Semi-Implicit Method for Pressure-Linked Equations Con- sistent, konsistentti puoli-implisiittinen paineenratkaisualgoritmi
TVD-ehto	engl. Total Variation Diminishing condition, totaali variaation pie- nemisehto
UD	engl. Upwind scheme, ylävirran menetelmä
A	pinta-ala
a	malliongelman mielivaltainen kerroin
a_1	k-omega SST -mallin kerroin
a_C, a_F	diagonaali- ja naapurikerroin
b_C	vakiokerroin
C	elementti
C_0, C_1, C_2	virhearvion kerroin
$CD_{k\omega}$	ristidiffuusiokerroin
c	malliongelman mielivaltainen kerroin
D	Taylorin sarjakehitelmän kerroin
D_f	geometrinen diffuusiokerroin
\mathbf{d}	etäisyysvektori
d	etäisyysvektorin pituus
\mathbf{E}_f	pinnan normaalin suuntainen pintavektori
E	paineen diskretisointivirhe
\mathbf{e}	kontrollitilavuuksia yhdistävän suoran suuntainen yksikkövektori
\mathbf{e}_u	nopeusvektorin diskretisointivirhe

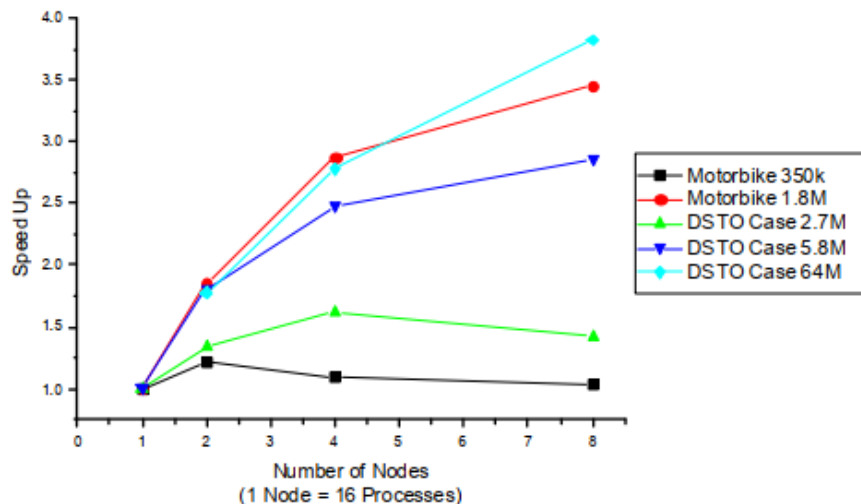
e	diskretisointivirhe
F_1, F_2	k-omega SST -mallin sekoitusfunktiot
F_T, F_V, F_E	aikakeskiarvo, tilavuuskeskiarvo ja koekeskiarvo
$F^{\phi,D}, F^{\phi,C}, F^{\phi,Q}$	diffuusiovuon, konvektiovuon ja lähdetermin normalisointikerroin
F^ϕ	jäännösmentelmän normalisointikerroin
\mathbf{f}	Navierin–Stokesin yhtälöiden lähde-termi
f	kontrollitilavuuden pinta
\hat{f}	malliongelman funktio
G_C	minimoitava gradienttiyhtälö
G_h	gradienttioperaattori
\mathbf{g}	gravitaatiovektori
g	interpolatiokerroin
\hat{g}	malliongelman gradienttireunaehto
H	Taylorin sarjakehitelmän korkeamman asteen termit
h	laskentaverkon koko
k	turbulenttinen kineettinen energia
L	1D laskentaverkon itä- ja länsinaapurin välinen etäisyys
\mathbf{m}_f	kontrollitilavuuksien keskipisteiden välinen puolikas etäisyysvektori
\dot{m}_f	pinnan massavirta
m^ϕ, m^u	skalaarimuuttujan ja nopeusvektorin toinen momentti
\mathbf{n}	pinnan normaalin suuntainen yksikkövektori
n	iteraatiokierros
n_k	yksikkövektorin mielivaltainen kerroin
P	paine
Pe_L	Pecletin luku
\widetilde{P}_k	turbulenssia rajoittava funktio
p, p_h, p_0	ominaispaine, diskretisoitu ominaispaine ja staattinen ominaispaine
p'	paineenkorjaus
Q	monikulmio
Q^ϕ	skalaarimuuttujan lähde-termi
q	painofunktion potenssi
$\mathbf{R}_v, \mathbf{R}_C$	oleellisten ja luonnollisten reunaehto- jen vuo-jäännös
R	vuo-jäännös
\mathbf{r}_C	Navierin–Stokesin yhtälöiden jäännösvektori
r_k	peräkkäisten gradienttien suhde
r	jäännös
$\bar{r}, \bar{r}_m, \bar{r}_r$	jäännöksen, momenttijäännöksen ja jäännösvirheen tilavuus- integraali
\mathbf{S}	pinnan normaalin suuntainen pintavektori
S_{II}	muodonmuutosnopeustensorin invariantti
S_f	kontrollitilavuuden pinnan pinta-ala
S_{ij}	keskimääräinen venymänopeustensori
s	diskretisointimenetelmän kertaluku
\mathbf{T}_f	epäortogonaalinen pintavektori
T, T_{eff}	keskiarvoistamiseen käytetty aika ja ominaisaikaskaala
TV	totaalivariaatio
$t, \Delta t$	aika ja aika-askel
\mathbf{U}	todellinen nopeusvektori

\mathbf{u}, \mathbf{u}_h	ajallisesti keskiarvoistettu ja diskretisoitu nopeusvektori
u'_i	hetkellinen nopeusvaihtelu
u_{eff}	ominaisnopeusskaala
u_h	diskretisoitu nopeusvektori
\bar{u}_i	nopeusvektorin väliaikainen arvo
$V, V_c, \partial V_c$	tilavuus, kontrollitilavuus ja kontrollitilavuuden reuna
w_k	painokerroin
\mathbf{x}	paikkavektori
x_k	x-suuntainen paikka
y	seinämäetäisyys
y_k	y-suuntainen paikka
z_k	z-suuntainen paikka
α_c	tasapainotettujen jäännösten virhe-estimaatin interpolointikerroin
$\alpha^U, \alpha^p, \alpha^\phi$	nopeuden, ominaispaineen ja ϕ -muuttujan alirelaksointikertoimet
$\beta, \beta_1, \beta_2, \beta^*$	k-omega SST -mallin kerroin
β_m	Gamma-menetelmän rajoitusparametri
Γ^ϕ	skalaarimuuttujan diffuusiokerroin
Γ_N	luonnollisia reunaehtoja sisältävä reuna
Γ_D	oleellisia reunaehtoja sisältävä reuna
γ	elementin tai kontrollitilavuuden reuna
$\gamma_\omega, \gamma_{\omega 1}, \gamma_{\omega 2}$	k-omega SST -mallin kerroin
γ_m	Gamma-menetelmän sekoituskerroin
ϵ	turbulenttisen kineettisen energian dissipaatio
ϵ^ϕ	muutosuhde
η, η_E, η_G	virhe-estimaatti, eksplisiittinen ja gradienttinvirhe-estimaatti
η_{GCI}, η_R	laskentaverkon suppenemisindeksi ja Richardsonin virhe-estimaatti
η_m, η_r	momenttimenetelmän ja jäännösvirhemenetelmän virhe-estimaatti
Θ	Swebyn vuorajoittimen parametri
θ	epäortogonaalisuus
ϑ	epäkeskeisyys
λ	kahden laskentaverkon kokojen suhde
μ, μ_t	molekulaarinen viskositeetti ja turbulenttinen viskositeetti
ν, ν_t	kinemaattinen viskositeetti ja pyörreviskositeetti
ξ	lokaalin jäännösongelman virhemuuttujavektori
ξ	lokaalin jäännösongelman virhemuuttuja
Π_s	interpolointioperaattori
ρ	tiheys
$\sigma_\omega, \sigma_{\omega 1}, \sigma_{\omega 2}$	k-omega SST -mallin kerroin
$\sigma_k, \sigma_{k1}, \sigma_{k2}$	k-omega SST -mallin kerroin
τ	Reynoldsin jännitystensori
Φ	Muuttujan tarkka arvo
$\phi, \tilde{\phi}, \bar{\phi}$	mielivaltainen, normalisoitu ja väliaikainen skalaarimuuttuja
φ	implisiittinen virhe-estimaatti
ψ	vuorajoitin
$\Omega, \partial\Omega$	laskenta-alue ja laskenta-alueen reuna
Ω_ω	pyörteisyyden suuruus
ω	ominaisdissipaatio
ω_{ip}	Gaussin integrointipiste

1. JOHDANTO

1.1 Työn tausta, tutkimuskysymykset ja tutkimusmenetelmät

Diplomityötä lähetettiin tekemään siitä lähtökohdasta, että valitettavan usein virtauslaskennan simulaatiotuloksissa esiintyy laskennasta johtuvaa diskretisointivirhettä, jonka suuruutta on objektiivisesti vaikea arvioida. Tyypillisesti laskennassa käytetty laskentateho ei salli simulaatiotulosten laskemista mielivaltaisella tarkkuudella erityisesti käytännön insinöörisovelluksissa, eikä tulosten diskretisointivirheen taso ole itsestään selvä. Toisinaan virtauslaskijalle tulee vastaan seuraava ongelma: vaikka virtauslaskenta on tehty moitteettomasti alusta lähtien ja parhaita virtauslaskentakirjallisuuden käytänteitä noudattaen (Ferziger & Perić 2002, Moukalled et al. 2016), ei numeerinen ratkaisu silti vastaa käytännön kokeissa saatuja tuloksia. Jos laskennassa käytettyjen yhtälöiden ja mallien oletetaan kuvaavan fysikaalisia ilmiöitä riittävän hyvin, ja kaikki saatavilla olevat resurssit, kuten esimerkiksi laskentateho on hyödynnetty, niin miten tulosten tarkkuutta voidaan enää näissä tapauksissa parantaa? Joudutaanko vain hyväksymään saavutettu virhetaso?



Kuva 1.1 Virtauslaskennan (OpenFOAM) skaalaavuus kaksi Intel XEON-E5 2687W -prosessoria sisältävillä työasemilla (Keough 2014, s. 19)

Eräänä keinona simulaatiotuloksissa esiintyvän diskretisointivirheen pienentämiseen on laittaa useita tietokoneita laskemaan yhtä virtausmallia kuvan 1.1 kaltaisesti, jolloin saatavilla olevat resurssit kasvavat ja laskentaverkkoa voidaan merkittävästi tihentää. Tulosten virhetaso tällöin paranee ja jäljelle jääneet virheet voivat olla sovelluskohteesta riippuen hyväksyttävän pieniä. Ongelmana tässä ratkaisutavassa on se, että esimerkiksi Keough (2014, s. 1–20) mukaan virtauslaskentasovelluksiin (OpenFOAM) laskentatehoa

ei voida rajatta lisätä HPC-teknologialla (engl. High Performance Computing), vaan saavutettu lisähyöty katoaa todella nopeasti, sillä useampien tietokoneiden tuoma monimutkaisuus vähentää laskennan hyötysuhdetta. Tällöin laskentateho saturoituu kuvan 1.1 osoittamalla tavalla, eikä tulosten tarkkuus enää tietyn pisteen jälkeen huomattavasti parane. Vaikka tulosten tarkkuutta todennäköisesti saadaan parannettua merkittävästi, niiden absoluuttisesta virhetasosta ei kuitenkaan ole tietoa. Tästä syystä tätä tapaa ei voida pitää hyvänä pidemmän aikavälin ratkaisuna, vaan on mietittävä vaihtoehtoisia ratkaisuja.

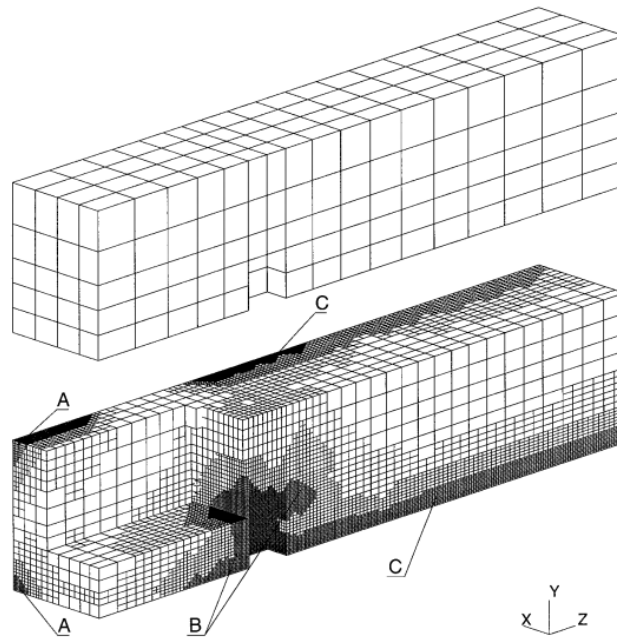
Toisena, ja ehdottomasti realistisempänä tapana, on parantaa virtauslaskennassa käytetyn virtausmallin tarkkuutta lisäämällä laskentaverkkoon resoluutiota vain niihin paikkoihin, joissa sitä todella tarvitaan. Tällöin saadaan suhteellisesti tarkempia tuloksia resursseihin nähden, koska laskentatehoa hukataan vähemmän. Haasteena on se, että miten lasketuista simulaatiotuloksista saadaan esille niiden virhetaso ja toisaalta millä logiikalla laskentaverkkoa tihennetään, jotta laskentaverkon tiheyttä muutetaan oikeissa paikoissa. Objektiiivinen virhearvio ja tämän perusteella hienonnettu laskentaverkko ei tietenkään takaa avaimia menestykseen vaan on varmistuttava siitä, että valitut laskentamenetelmät ovat parhaat mahdolliset, eli virhetasot ovat jo alkujaankin mahdollisimman pieniä, erityisesti käytännön insinöörisovelluksissa. Tällöin lopputulokset voivat olla hyviä, vaikka käytettäisiinkin vain kohtuullista laskentatehoa, kuten esimerkiksi tässä työssä hyödynnettyä Intel i5-4690k -prossessorin ja 16 GB -keskusmuistin sisältävää tietokonetta.

Edellä mainittujen seikkojen vuoksi tässä diplomityössä pyritään vastaamaan seuraavansiin tutkimuskysymyksiin:

1. Miten kontrollitulavuusmenetelmällä saavutetaan toisen kertaluvun diskretisointivirheen pienemisenopeus, joka on mahdollisimman riippumaton käytetystä laskentaverkon tyypistä?
2. Miten saadaan muodostettua kontrollitulavuusmenetelmällä turbulenttisten ja koonpuristumattomien Navierin–Stokesin yhtälöiden virhearvio ja tämän avulla arvioitua diskretisointivirheen ylärajaa luotettavasti?
3. Miten virhearvioon perustuva adaptiivinen laskentaverkon tihentäminen parantaa numeerisen ratkaisun tarkkuutta perinteiseen rajakerroksia tarkemmin mallintavaan menetelmään verrattuna?
4. Korreloiko virhearviot myös kokeellisen datan kanssa?

Ensimmäistä tutkimuskysymystä pohditaan työn toisessa luvussa, kun syvennyttään virtauslaskennassa käytetyn kontrollitulavuusmenetelmän periaatteisiin (Moukalled et al. 2016), erityisesti siinä käytettyihin konvektiomenetelmiin (Sweby 1984, Jasak 1996), gradienttimenetelmiin (Syrakos et al. 2017) sekä pikaisesti myös aikaintegrointimenetelmiin (Ferziger & Perić 2002). Toiseen tutkimuskysymykseen paneudutaan tarkemmin kolmannessa luvussa käymällä läpi jälkikäteisvirhearvioiden teoriapohjaa (Ainsworth & Oden 1997) ja samalla tuodaan esille virtauslaskennan tarpeisiin kehitettyjä virhearviomenetelmiä, kuten momenttinvirhe- ja jäännösvirhe-estimaatit (Jasak 1996) sekä sivutaan Richardsonin ekstrapolointiin pohjautuvia menetelmiä (Roache 1994). Diplomityön nel-

jännessä luvussa käsitellään kuvan 1.2 kaltaiset adaptiiviset laskentaverkot mahdollistavaa algoritmia sekä sen toteuttamista OpenFOAM-sovellukseen. Toteutetun adaptiivisen algoritmin avulla tuotetaan myöhemmin simulaatiotuloksia ja tuloksia analysoidaan viidennessä luvussa, ja tulosten perusteella vastataan kolmanteen tutkimuskysymykseen. Lopuksi työssä vielä vertaillaan virhearvioiden ennustamaa virhetasoa kokeelliseen dataan nähden ja vastataan viimeiseen tutkimuskysymykseen. Diplomityön tutkimusmenetelminä ovat siis teoriakatsaus, simulointi ja käytännön koe.



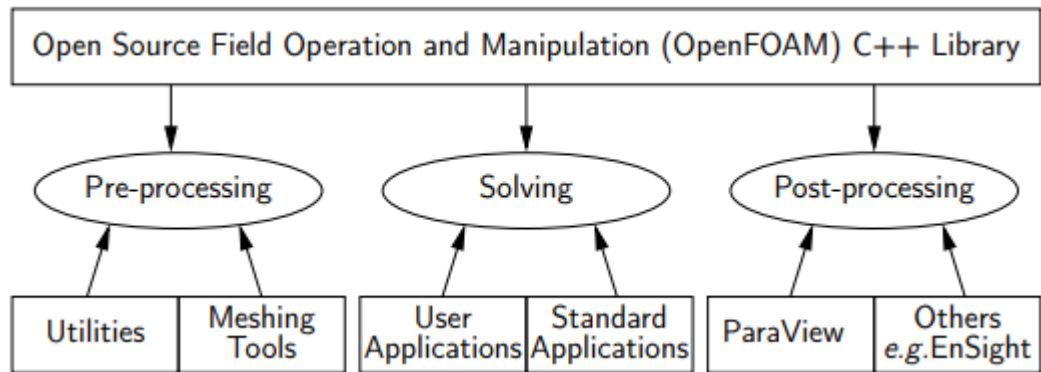
Kuva 1.2 Virtauslaskennassa käytetty h-adaptiivinen laskentaverkko (Muzaferija & Gosman 1997, s. 782)

Yleisesti kirjallisuudessa toteutettava algoritmi tunnetaan h-adaptiivisuutena, jonka avulla laskentaverkon lokaalia kokoa muokataan virhearvion määrittelemissä kohdissa ja uusi laskentaverkko perustuu näin aina edelliseen simulaatiotulokseen. Vaihtoehtoisesti diskretisointimenetelmien tarkkuutta voitaisiin nostaa joko paikoitellen tai koko laskentaverkossa (p-adaptiivisuus) tai jakaa olemassa olevan laskentaverkon solmut uudelleen siten, että niistä saadaan eniten hyötyä (r-adaptiivisuus). Eri menetelmistä h-adaptiivisuus on ehdottomasti suosituin. (Zienkiewicz et al. 2013, s. 545–548) Kyseisen h-adaptiivisuuden erinomaisen soveltuvuuden virtauslaskentaan ovat todistaneet esimerkiksi Jasak (1996) ja Muzaferija & Gosman (1997).

1.2 OpenFOAM-virtauslaskentasovellus

Ennen teoriaosuuteen siirtymistä tutustutaan lyhyesti työhön valittuun virtauslaskentasovellukseen, sillä simulointi kattaa ison osan koko työstä. Näin saadaan muutamia tarvittavia perustietoja, ennen kuin lisäominaisuuksia lähdetään myöhemmin toteuttamaan, sillä kirjoittajan tiedossa ei ole täysin valmiita ratkaisuja, joita voisi suoraan hyödyntää.

Työn simulaatio-osuus suoritetaan avoimeen lähdekoodiin pohjautuvalla OpenFOAM v5.0 -sovelluksella (engl. Open Source Field Operation and Manipulation), joka on koelma erilaisia virtauslaskennassa tarvittavia esikäsittelyyn, ratkaisemiseen ja jälkikäsittelyyn liittyviä sovelluksia (Greenshields 2017, s. 17). Erityisen sopivan sovelluksesta diplomityön puitteissa tekee kuvassa 1.3 esitetty sovellusrakenne, joka mahdollistaa käyttäjän tekemien sovellusten ajamisen. Käytännössä tämä tarkoittaa, että adaptiivisen algoritmin toteuttamisen työmäärä on hieman pienempi, sillä ohjelmointitekniisiin seikkoihin ei tarvitse käyttää niin paljoa aikaa.



Kuva 1.3 OpenFOAM-virtauslaskentasovelluksen rakenne (Greenshields 2017, s. 17)

Sovelluksen perustoiminnallisuus pohjautuu aiemmin kehitettyihin FOAM-kirjastoihin (engl. Field Operation and Manipulation), joiden erityispiirteenä on niiden suoraviivainen mallinnuskieli, sillä sovelluksessa käytetty koodi heijastaa ratkaistavien differentiaaliyhtälöiden muotoa. Tämän vuoksi kirjastot sopivat hyvin myös muiden mekaniikan osalueiden mallintamiseen, kuten esimerkiksi lujuslaskentaan tai magnetohydrodynamiikkaan. Ratkaistavien yhtälöiden diskretisointi on kohtuullisen vaivatonta, sillä esimerkiksi jatkuvuusyhtälön $\partial\rho/\partial t + \nabla \cdot \phi = 0$ ohjelmointi onnistuu yksinkertaisella koodilla

```

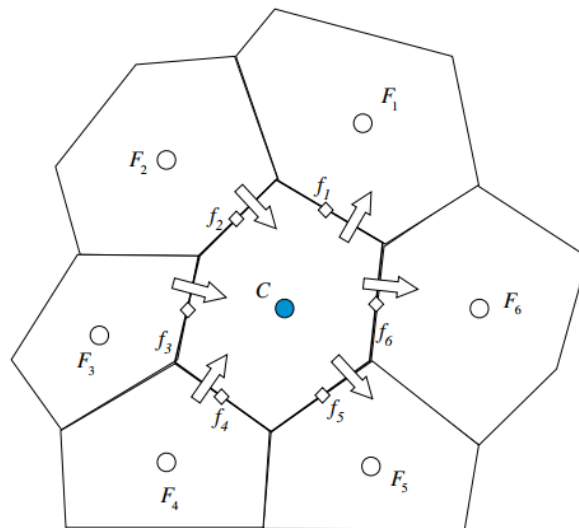
fvMatrixScalar rhoEq
(
    fvm::ddt(rho)+fvc::div(phi)
);
  
```

ja yhtälö voidaan myöhemmin ratkaista suoraviivaisesti käskyllä `rhoEq.solve()`. FOAM-mallinnuskielessä funktioiden edellä oleva `fvm`-etuliite merkitsee sitä, että termistä muodostetaan matriisi valittujen diskretisointimenetelmien avulla ja `fvc`-etuliite puolestaan sitä, että suuresta muodostetaan skalaari-, vektori- tai tensorimuotoinen kenttä. Näin käyttäjä voi valita, että minkä termin suhteen yhtälö halutaan ratkaista. Sovellus sallii myös matemaattisten operaattoreiden käyttämisen sellaisenaan kuten esimerkiksi yhtälöissä esiintyvien termien summaamisen tai näiden kertomisen keskenään. Myös termien yksiköt on otettu huomioon, sillä sovellus automaattisesti tarkastaa, vastaavatko laskettavien termien yksiköt toisiaan, ja ilmoittaa mikäli näin ei käy. Weller et al. (1998, s. 620–630) Tarkemmin OpenFOAM:in ohjelmointitekniisiin seikkoihin paneudutaan myöhemmin työn neljännessä luvussa.

2. DIFFERENTIAALIYHTÄLÖIDEN DISKRETISOINTI KONTROLLITILAVUUSMENETELMÄLLÄ

2.1 Kontrollitilavuusmenetelmän periaate

Lähtökohtaisesti kontrollitilavuusmenetelmässä mielenkiinnonkohteena oleva laskenta-alue jaetaan pienempiin diskreetteihin osatilavuuksiin siten, että koko alue täyttyy näistä kontrollitilavuuksista. Kontrollitilavuuksien keskipisteeseen sijoitetaan laskentasolmu, jonka oletetaan kuvaavan toisen kertaluvun tarkkuudella kontrollitilavuuden keskimääräistä arvoa. Vaihtoehtoisesti laskenta-alueelle asetetaan jokin määrä laskentasolmuja ja rakennetaan kontrollitilavuudet näiden solmujen ympärille siten, että kontrollitilavuuksien väliset pinnat ovat yhtä kaukana solmukohdista. Erona tässä menetelmässä on se, että pintaintegraalit määritetään luonnostaan tarkemmin, mutta muuttujan arvoa ei enää välttämättä kuvata toisen kertaluvun tarkkuudella kontrollitilavuuden sisällä. (Ferziger & Perić 2002, s. 71–72) Yleisesti kontrollitilavuusmenetelmän hyötyinä voidaan pitää sen joustavuutta ja erityisesti kykyä säilyttää ratkaistavien differentiaaliyhtälöiden integraaliominaisuus. Toisaalta kontrollitilavuusmenetelmä on helppokäyttöinen, koska erillisiä laskentakoordinaatiston muunnoksia ei tarvita. (Moukalled et al. 2016, s. 103–104)



Kuva 2.1 Tyypillinen laskentaverkko, yksittäisen kontrollitilavuuden säilymisominaisuus sekä kontrollitilavuutta ympäröivät naapurit (Moukalled et al. 2016, s. 109)

Paneudutaan tässä luvussa tarkemmin siihen, miksi kuvan 2.1 kaltainen, mielivaltaisista monikulmioista muodostettu laskentaverkko, ei aina johda tarkkuuden kannalta hyväksyttävään lopputulokseen ja siihen, miten differentiaaliyhtälöt ylipäätään ratkaistaan kontrollitilavuusmenetelmällä. Helpoiten tämä käy tutkimalla esimerkkiyhtälöä, jossa on monia todellisten virtauskenttä mallintavien yhtälöiden piirteitä.

Perinteisesti kontrollitilavuusmenetelmässä lähdetään diskretisoimaan säilymsyhtälöä

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\mathbf{u}\phi) = \nabla \cdot (\Gamma^\phi \nabla \phi) + Q^\phi, \quad (2.1)$$

jossa ϕ on jokin skalaarimuuttuja, ρ tiheys ja Γ^ϕ diffuusiokerroin. Integroimalla kontrollitilavuuden V_c yli ja jättämällä pois yhtälön (2.1) vasemman puolen aikatermi, saadaan ajasta riippumattomaksi integraalimuotoiseksi yhtälöksi

$$\int_{V_c} \nabla \cdot (\rho\mathbf{u}\phi) dV = \int_{V_c} \nabla \cdot (\Gamma^\phi \nabla \phi) dV + \int_{V_c} Q^\phi dV, \quad (2.2)$$

jonka tarkka ratkaisu takaa, että kaikkien kontrollitilavuuteen sisään- ja ulostulevat vuot ovat yhtäsuuret ja häviöttömät. Divergenssiteoreeman nojalla on mahdollista yksinkertaistaa yhtälössä (2.2) esiintyviä integraaleja, minkä seurauksena yhtälölle saadaan muoto

$$\oint_{\partial V_c} (\rho\mathbf{u}\phi) d\mathbf{S} = \oint_{\partial V_c} (\Gamma^\phi \nabla \phi) d\mathbf{S} + \int_{V_c} Q^\phi dV, \quad (2.3)$$

jossa \mathbf{u} on nopeusvektori, \mathbf{S} pinnan normaalin suuntainen pintavektori ja Q^ϕ lähde-termi. (Moukalled et al. 2016, s. 104–105) Mikäli kontrollitilavuusmenetelmän halutaan olevan toisen kertaluvun menetelmä mielivaltaisen muotoisissa kontrollitilavuuksissa, tulee kaikkien yhtälön (2.3) diskretisointimenetelmien toteuttaa tämä vaatimus. Yhtälön (2.3) pinta- ja tilavuusintegraalien diskretisointi asettaa puitteet täten koko numeerisen ratkaisun tarkkuudelle.

Integroimalla kontrollitilavuuden pintaintegraalit jokaisen pinnan yli, saadaan konvektio-termin sisältävä pintaintegraali muutettua summamuotoon

$$\oint_{\partial V_c} (\rho\mathbf{u}\phi) d\mathbf{S} = \sum_{f=1}^{n(f)} \left(\int_f (\rho\mathbf{u}\phi) \cdot d\mathbf{S} \right), \quad (2.5)$$

ja vastaavasti saadaan myös diffuusiotermin sisältävä integraali muotoon

$$\oint_{\partial V_c} (\Gamma^\phi \nabla \phi) d\mathbf{S} = \sum_{f=1}^{n(f)} \left(\int_f (\Gamma^\phi \nabla \phi) \cdot d\mathbf{S} \right). \quad (2.6)$$

Integraalien (2.5) ja (2.6) tarkka integrointi säilyttää kontrollitilavuusmenetelmän konservatiivisuuden. Jotta diskretisoinnissa päästään eteenpäin, tarvitaan keino laskea konvektio- ja diffuusiovuon sisältävien yksittäisten pintojen integraalit ilman niiden tarkkaa ratkaisemista. Gaussin kvadratuurin mukaisesti pinta-integraaleille (2.5) ja (2.6) voidaan johtaa tulokset

$$\oint_{\partial V_c} (\rho \mathbf{u} \phi) d\mathbf{S} = \sum_{f=1}^{n(f)} \sum_{ip=1}^{n(ip)} [\omega_{ip} (\rho \mathbf{u} \phi)_{ip} \cdot \mathbf{S}_f], \quad (2.7)$$

ja

$$\oint_{\partial V_c} (\Gamma^\phi \nabla \phi) d\mathbf{S} = \sum_{f=1}^{n(f)} \sum_{ip=1}^{n(ip)} [\omega_{ip} (\Gamma^\phi \nabla \phi)_{ip} \cdot \mathbf{S}_f], \quad (2.8)$$

jossa ω_{ip} merkitsee painofunktion arvoa integrointipisteessä ip ja \mathbf{S}_f pinnan normaalin suuntaista vektoria, jonka suuruutena käytetään kyseisen pinnan f pinta-alaa S_f . Jos pintaintegraalit (2.7) ja (2.8) halutaan määrittää toisen kertaluvun tarkkuudella, riittää, että käytetään vain yhtä integrointipistettä ja asetetaan painofunktiolle $\omega_{ip} = 1$. Tässä tapauksessa integraalin arvo määräytyy pinnan f keskipisteessä tunnetun arvon perusteella. Korkeamman asteen kontrollitilavuusmenetelmiä voidaan muodostaa, kun lisätään integrointipisteiden määrää. (Moukalled et al. 2016, s. 105–107)

Lähdetermin tapauksessa tilavuusintegraali saadaan muodostettua vastaavalla tavalla kuin edellä asettamalla useita Gaussin integroimispisteitä kontrollitilavuuden sisälle. Tällöin lähdetermin tilavuusintegraali voidaan määrittää yhteydestä

$$\int_{V_c} Q^\phi dV = \sum_{ip=1}^{n(ip)} (Q_{ip}^\phi \omega_{ip} V), \quad (2.9)$$

jossa toisen kertaluvun tarkkuuteen riittää, kun kontrollitilavuuden keskipisteeseen sijoitetaan vain yksi Gaussin integroimispiste. Tällöin valitaan painofunktioksi $\omega_{ip} = 1$, eli lähdetermin suuruus arvioidaan vain kontrollitilavuuden keskipisteen arvon perusteella. Soveltamalla tuloksia (2.7-2.9) yhtälöön (2.3) saadaan kontrollitilavuuskohtainen yhtälö

$$\sum_{f=1}^{n(f)} (\rho \mathbf{u} \phi - \Gamma^\phi \nabla \phi)_f \cdot \mathbf{S}_f = Q_C^\phi V_C, \quad (2.10)$$

jota yleensä kutsutaan puolidiskretisoiduksi yhtälöksi, koska yhtälössä esiintyvät vuot ja tilavuuskohtaiset arvot eivät ole vielä tunnettuja. (Moukalled et al. 2016, s. 107–108)

Luvuissa 2.3-2.5 käsitellään tarkemmin sitä, miten yhtälössä (2.10) tarvittavat arvot määritetään toisen kertaluvun menetelmillä. Kontrollitilavuusmenetelmän tarkkuuden parantaminen isommalla määrällä integroimispisteitä on varsin houkuttelevaa, mutta yhtälön (2.10) termien arvioiminen toisen kertaluvun tarkkuudella on itsessään jo suuri haaste, eikä tämän toteutumisesta välttämättä ole varmuutta mielivaltaisen muotoisten kontrollitilavuuksien tapauksessa kuten myöhemmin osoittautuu. Tarkempi integrointi ei välttämättä tuo lisähyötyä, joten pyritään toisen kertaluvun tarkkuuteen tässä työssä.

2.2 Ratkaistavat yhtälöt

Ennen yksityiskohtaisempien diskretisointimenetelmien johtamista on paikallaan määrittellä yhtälöt, joille työssä haetaan numeerista ratkaisua. Olettamalla virtauksen olevan kokoonpuristumatonta, virtausta voidaan kuvata tarkasti massan säilyvyysyhtälön

$$\frac{\partial U_i}{\partial x_i} = 0, \quad (2.11)$$

ja liikemäärän säilymisyhtälön

$$\rho \frac{\partial U_i}{\partial t} + \rho U_j \frac{\partial U_i}{\partial x_j} = \rho g_i - \frac{\partial P}{\partial x_i} + \mu \frac{\partial^2 U_i}{\partial x_j^2}, \quad (2.12)$$

avulla, joista yhtälöä (2.12) yleensä kutsutaan Navierin–Stokesin yhtälöiksi (Nieuwstadt et al. 2016, s. 10–11). Liiketyhtälössä termi g_i kuvaa gravitaatiovoiman i -suuntaista komponenttia ja μ dynaamista viskositeettia. Yhtälöiden ratkaiseminen insinöörisovelluksissa ei ole Nieuwstadt et al. (2016, s. 71–74) mukaan järkevää ilman yhtälöiden merkittävää yksinkertaistamista turbulenssimallilla, koska yhtälöiden vaatima resoluutio rajoittaa yhtälöiden käyttämisen lähinnä laminaarisille virtauksille. Työssä ratkaistaan pääasiassa turbulenttisia virtauksia, joten ongelma kierretään muuttamalla Navierin–Stokesin yhtälöt ajallisesti keskiarvoistetuiksi yhtälöiksi. Wilcoxin (1993, s. 16) mukaan virtausmekaniikassa kyseiset yhtälöt yleensä tunnetaan nimellä RANS-yhtälöt eli Reynoldsin keskiarvoistetut Navierin–Stokesin yhtälöt (engl. Reynolds Averaged Navier–Stokes equations).

Navierin–Stokesin yhtälöiden ajallinen keskiarvoistaminen hyödyntää nimensä mukaisesti Reynoldsin esittämää keskiarvomenetelmää. Yleisellä tasolla ajasta riippumattoman muuttujan $\hat{f}(\mathbf{x}, t)$ aikakeskiarvo $F_T(\mathbf{x})$ voidaan määrittää yhteydestä

$$F_T(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \hat{f}(\mathbf{x}, t) dt, \quad (2.13)$$

jossa T on mallinnettavien ilmiöiden kannalta riittävän pitkä integrointiaika. Toisena vaihtoehtona on määrittää muuttujan $f(\mathbf{x}, t)$ tilavuuskeskiarvo integroimalla se tilavuuden V yli, eli

$$F_V(t) = \lim_{V \rightarrow \infty} \frac{1}{V} \iiint \hat{f}(\mathbf{x}, t) dV. \quad (2.14)$$

Kolmantena mahdollisuutena on muodostaa kokeisiin perustuva koekeskiarvo summamalla tulokset $\hat{f}_n(\mathbf{x}, t)$ ja jakamalla ne kokeiden lukumäärällä, jolloin saadaan koekeskiarvoksi määritettyä

$$F_E(\mathbf{x}, t) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \hat{f}_n(\mathbf{x}, t), \quad (2.15)$$

joka ei aseta erityisehtoja keskiarvon käyttämiselle. Sen sijaan ajan suhteen keskiarvoistamisessa ja tilavuuskeskiarvon laskemisessa täytyy varmistua, voidaanko näitä ylipääntään soveltaa. Koska keskiarvomenetelmiä (2.13–2.15) sovelletaan turbulentsisiin virtauksiin, vain (2.15) on yleisesti voimassa. Mikäli turbulenssi on stationääristä, eli ajasta riippumattonta, voidaan hyödyntää aikakeskiarvoa. Jos taas turbulenssi on homogeenista, toisin sanoen tasaisesti jakautunutta spatiaalisesti, on mahdollista määrittää tilavuuskeskiarvo. Useimmissa virtaustapauksissa vain edellä mainittu stationäärisyysehto täyttyy, joten aikakeskiarvoa voidaan helpoimmin hyödyntää Navierin–Stokesin yhtälöihin, jos yhtälöillä halutaan kuvata virtauskentän keskiarvosuureita, eikä yksittäisiä pyörteitä välttämättä ole tarkoitus ratkaista. (Wilcox 1993, s. 11–13)

Turbulenssin käsittely on itsessään oma laaja tieteenalansa, joten oletetaan aikakeskiarvojen kuvaavan riittävällä tarkkuudella virtauskentän keskiarvosuureita, mikä mahdollistaa tässä työssä esiintyvien virtaustapauksien ratkaisemisen. Vaihtoehtoinen suurten pyörteiden menetelmä, eli LES-menetelmä (engl. Large Eddy Simulation), löytyy kiinnostuneelle lukijalle esimerkiksi lähteestä (Pope 2000), kompakti katsaus turbulenssi-ilmiöihin lähteestä (Nieuwstadt et al. 2016) ja suuren joukon erilaisia turbulenssimalleja esittää esimerkiksi (Wilcox 1993).

Yksinkertainen tapa muodostaa aikakeskiarvo stationäärille muuttujalle on jakaa se keskiarvo-osaan ja ajallisesti vaihtelevaan osaan. Nopeusvektori \mathbf{U} voidaan esittää muodossa

$$U_i(\mathbf{x}, t) = u_i(\mathbf{x}) + u_i'(\mathbf{x}, t), \quad (2.16)$$

jossa esiintyvä keskiarvo $u_i(\mathbf{x})$ voidaan määrittää aikakeskiarvon

$$u_i(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} U_i(\mathbf{x}, t) dt \quad (2.17)$$

avulla. Keskiarvo on voimassa, mikäli integrointiaika T on merkittävästi suurempi kuin turbulentsisessa virtauksessa esiintyvä pisin aikaskaala. Hyödyntämällä määritettyä aikakeskiarvoa saadaan kokoonpuristumattomien Navierin–Stokesin yhtälöiden ja jatkuvuusyhtälön aikakeskiarvoistetuiksi muodoiksi

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (2.18)$$

ja

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (2\mu S_{ji} - \rho \overline{u_j' u_i'}), \quad (2.19)$$

jossa esiintyvä tensori S_{ij} kuvaa muodonmuutosnopeutta, ja tämä saadaan yhteydestä

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (2.20)$$

Uutena terminä ilmestynyttä korrelaatiotermiä $\tau_{ij} = -\rho \overline{u'_i u'_j}$ yleisesti kutsutaan Reynoldsin jännitystensoriksi, joka on symmetrinen. (Wilcox 1993, s. 15–16) Liikemäärän säilymisyhtälöiden (2.12) sisältämä gravitaatiotermi on jätetty pois yhtälöistä (2.19)

Yhtälö (2.19) kirjoitetaan jatkoa varten hieman toisenlaiseen muotoon jakamalla molemmat puolet tiheydellä ρ , merkitsemällä $p = P/\rho$ ja muuttamalla viskositeetti μ kinemaattiseksi viskositeetiksi yhteyden $\nu = \mu/\rho$ avulla. Työssä ratkaistavaksi yhtälöksi näin ollen saadaan

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} (2\nu S_{ji} - \overline{u'_j u'_i}), \quad (2.21)$$

ja jatkossa keskimääräistä nopeutta \mathbf{u} kutsutaan lyhennetyksi nopeudeksi ja ominaispainetta p lyhyesti paineeksi.

Jännitystensorille $\boldsymbol{\tau}$ voidaan johtaa Reynoldsin keskiarvomenetelmällä erillinen differentiaaliyhtälö, ja lopputuloksena on kuusi uutta ratkaistavaa yhtälöä, mutta valitettavasti yhtälöitä ei voida ratkaista ilman sulkeumahypoteeseja. Huomattavasti suoraviivaisempaa on hyödyntää yleisesti tunnettua Boussinesqin hypoteesia ja muodostaa kokoonpuristumattomassa tapauksessa turbulenttisen viskositeetin μ_t avulla yhteys

$$\tau_{ij} = -\rho \overline{u'_i u'_j} = 2\mu_t S_{ij} - \frac{2}{3} \rho k \delta_{ij}, \quad (2.22)$$

jossa hyödynnetään turbulenssin kineettistä ominaisenergiaa k muodossa

$$k = \frac{1}{2} \overline{u'_i u'_i}, \quad (2.23)$$

ja tämä puolestaan voidaan muodostaa nopeusvaihteluiden u'_i avulla, jos nämä tunnetaan. (Wilcox 1993, s. 17–75)

Todellisuudessa nopeusvaihtelut u'_i ovat tuntemattomia, joten tarvitaan käyttökelpoisempi keino ratkaista k . Tämän vuoksi vuosikymmenien aikana on kehitetty laaja joukko erilaisia malleja tähän tarkoitukseen. Suosituimpana kahden yhtälön mallina voidaan pitää Launder & Sharma (1974, s. 131–133) kehittämää turbulenttisen kineettisen energian dissipaatioon ϵ perustuvaa k - ϵ -turbulenssimallia, jonka avulla k voidaan ratkaista. Mallin heikkoudet anisotrooppisissa turbulenssissa ja voimakkaassa pyörteilyssä ovat Nieuwstadt et al. (2016, s. 169–170) mukaan yleisesti tiedossa, ja malli voi muun muassa näissä tapauksissa aiheuttaa merkittävän suurin virheitä. Sen sijaan esimerkiksi Wilcoxin

(1993, s. 84–87) esittämä ominaisdissipaatioon ω pohjautuva k - ω -turbulenssimalli toimii hyvin näissä tilanteissa, mutta Menterin (1993, s. 2) mukaan malli on liian herkkä vapaan virtauksen turbulenssiarvoille eikä siten mallinna esimerkiksi leikkausjännitys-kerroksia oikein.

Parempana vaihtoehtona on yhdistää edellä mainitut turbulenssimallit siten, että rajakerroksissa ratkaistaan turbulenssi käyttäen k - ω -turbulenssimallia ja kauempana rajakerroksista k - ϵ -turbulenssimallilla. Näin päädytään kahteen uuteen turbulenssia kuvaavaan yhtälöön, joiden mukaisesti

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_T) \frac{\partial k}{\partial x_j} \right] \quad (2.24)$$

ja

$$\begin{aligned} \frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} &= \frac{\gamma_\omega}{\nu_t} \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] \\ &+ 2\rho(1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, \end{aligned} \quad (2.25)$$

missä F_1 -rajoitusfunktio varmistaa mallin noudattavan k - ω -turbulenssimallia lähellä seinämiä. Funktion arvo määräytyy seinämäetäisyyden y perusteella yhteyden

$$F_1 = \tanh \left\{ \left[\min \left(\max \left(\frac{k^{1/2}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\rho \sigma_{\omega 2} k}{CD_{k\omega} y^2} \right) \right]^4 \right\}, \quad (2.26)$$

mukaisesti, missä esiintyvän ristidiffuusiomuuttujan $CD_{k\omega}$ arvo saadaan yhtälöstä

$$CD_{k\omega} = \max \left(2\rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right). \quad (2.27)$$

Uusissa differentiaaliyhtälöissä esiintyvälle pyörreviskositeetille ν_t voidaan määrittää arvo pyörteisyyden itseisarvon Ω_ω avulla yhteydestä

$$\nu_t = \frac{a_1 k}{\max(a_1 \omega, \Omega_\omega F_2)}, \quad (2.28)$$

jossa hyödynnetään vastaavanlaista rajoitusfunktiota kuin aiemmin. Rajoitusfunktiona tässä tapauksessa käytetään

$$F_2 = \tanh \left\{ \left[\max \left(\frac{2k^{1/2}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right) \right]^2 \right\}, \quad (2.29)$$

ja tämän tavoitteena on varmistaa, että pyörreviskositeetti määräytyy pääasiassa yhteydestä $\nu_t = \mu_t/\rho = k/\omega$, ja F_2 -rajoitin kytkeytyy päälle, kun suhde k/ω kasvaa liian suureksi. Lopputuloksena olevaa mallia kutsutaan k - ω SST -turbulenssimalliksi. (Menter 1993, s. 2–5) Yhtälöiden sisältämät malliparametrit esitetään, kun kyseistä mallia sovelletaan myöhemmin viidennessä luvussa.

Turbulenssimalli on osoittautunut tutkimuskäytön lisäksi sopivan myös teollisuuden käyttötarpeisiin ja malli saadaan käytettävyyden kannalta perinteisen k - ϵ -turbulenssimallin tasolle, jos alkuperäiseen malliin tehdään pieniä muutoksia. Turbulenssimallista on mahdollista käyttää hieman päivitettyä formulaatiota, minkä mukaan yhtälössä (2.24) käytetään τ_{ij} sisältämän termin sijaan turbulenssin muodostamista rajoittavaa funktiota

$$\tau_{ij} \frac{\partial u_i}{\partial x_j} := \widetilde{P}_k = \min \left[\mu_t \frac{\partial u_i}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), 10\beta^* \rho k \omega \right], \quad (2.30)$$

ja pyörreviskositeetti määritetään uudesta yhteydestä

$$\nu_t = \frac{a_1 k}{\max(a_1 \omega, S_{II} F_2)}, \quad (2.31)$$

jossa käytetään nyt pyörteisyyden Ω_ω sijaan muodonmuutoksen nopeutta kuvaavan tensorin S_{ij} invarianttia S_{II} . (Menter et al. 2003, 625–627) Päivitetystä turbulenssimallissa ei sovellettu \widetilde{P}_k -rajoitinfunktiota yhtälön (2.25) vasemman puolen ensimmäiseen termiin, mutta Rumseyn (2015) mukaan kyseessä oli kuitenkin painovirhe, eli kyseinen termi päivitetään rajoitettuun muotoon

$$\frac{\gamma_\omega}{\nu_t} \tau_{ij} \frac{\partial u_i}{\partial x_j} := \frac{\gamma_\omega}{\nu_t} \widetilde{P}_k, \quad (2.32)$$

ja pyörreviskositeetille (2.31) tarvittava muodonmuutosnopeutta kuvaavan tensorin invariantti määritetään seuraavasti:

$$S_{II} = \sqrt{2S_{ij}S_{ij}}. \quad (2.33)$$

Nyt kaikki tarvittavat yhtälöt kokoonpuristumattomien ajallisesti keskiarvoistettujen Navierin–Stokesin yhtälöiden ratkaisemiseen on esitetty. Yhtälöiden (2.21) ja (2.24–2.25) voidaan havaita sisältävän monia luvussa 2.1 esitetyn skalaarimuuttujan säilymisyyhtälön piirteitä. Esimerkiksi yhtälössä (2.21) esiintyy aikaderivaatta $\partial u_i/\partial t$, gradientti $\partial p/\partial x_i$, divergenssi $\partial u_i/\partial x_i$ ja termin $\partial S_{ji}/\partial x_j$ vuoksi muodostuvat Laplacen termit. Muiden termien voidaan katsoa kuuluvan lähdetermiin. Keskitytään seuraavaksi Laplacen termiin, eli diffuusiotermien diskretisointiin.

2.3 Diffuusiotermi

Diffuusiotermin $\nabla \cdot (\Gamma^\phi \nabla \phi)$ diskretisointia on helpoin lähtöä muodostamaan, kun luvussa 2.1 esitetystä skalaarimuuttujan säilymisyyhtälöstä (2.1) eliminoidaan tarpeettomat termit ja jätetään vain diffuusiotermi ja lähde-termi jäljelle. Myös lukijalle on helpompaa seurata, kun termejä käsitellään yksitellen.

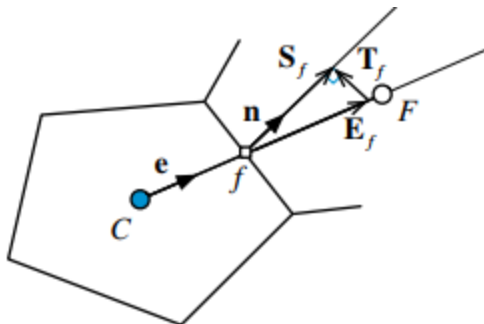
Stationäärisen diffuusioyhtälön

$$-\nabla \cdot (\Gamma^\phi \nabla \phi) = Q^\phi \quad (2.34)$$

kontrollitilavuusmenetelmän mukainen diskreetti muoto on

$$\sum_{f=1}^{n(f)} [-(\Gamma^\phi \nabla \phi)_f \cdot \mathbf{s}_f] = Q_C^\phi V_C, \quad (2.35)$$

jossa $n(f)$ merkitsee kontrollitilavuuden pintojen lukumäärää. Yhtälössä (2.35) esiintyvän gradientin arvioiminen yleisessä tapauksessa vaikeutuu, mikäli kontrollitilavuuden C ja sen naapurin F keskipisteitä yhdistävä vektori \mathbf{d}_{CF} ei ole samansuuntainen kuin pinnan normaalin suuntainen yksikkövektori \mathbf{n} . (Moukalled et al. 2016, s. 211–241)



Kuva 2.2 Kahden kontrollitilavuuden välisen pinnan epäortogonaalisuus ja tätä muoto-
virhettä kompensoiva ylirelaksoitu korjausmenetelmä (Moukalled et al. 2016, s. 243)

Kontrollitilavuuksien välillä esiintyvän pinnan epäortogonaalisuuden vuoksi diffuusioyhtälön diskretisoinnissa tarvittava pintavektori \mathbf{S}_f on jaettava pinnan normaalin suuntaiseen komponenttiin \mathbf{E}_f ja epäortogonaaliseen pintavektoriin \mathbf{T}_f yhteyden

$$\mathbf{S}_f = \mathbf{E}_f + \mathbf{T}_f \quad (2.36)$$

mukaisesti. Muodostamalla kontrollitilavuuksien keskipisteiden välisen suoran suuntainen yksikkövektori \mathbf{e} , joka tunnetaan geometrian perustella yhteydestä

$$\mathbf{e} = \frac{\mathbf{x}_F - \mathbf{x}_C}{\|\mathbf{x}_F - \mathbf{x}_C\|} = \frac{\mathbf{d}_{CF}}{d_{CF}}, \quad (2.37)$$

saadaan tämän vektorin suuntaiseksi gradientiksi määritettyä

$$(\nabla\phi \cdot \mathbf{e})_f = \left(\frac{\partial\phi}{\partial e} \right)_f = \frac{\phi_F - \phi_C}{\|\mathbf{x}_F - \mathbf{x}_C\|} = \frac{\phi_F - \phi_C}{d_{CF}}. \quad (2.38)$$

Gradientin (2.38) perusteella diffuusiovuoto voidaan kokonaisuudessaan, laskentaverkon epäortogonaalisuus huomioon ottaen, kirjoittaa muotoon

$$(\nabla\phi)_f \cdot \mathbf{S}_f = (\nabla\phi)_f \cdot \mathbf{E}_f + (\nabla\phi)_f \cdot \mathbf{T}_f = E_f \frac{\phi_F - \phi_C}{d_{CF}} + (\nabla\phi)_f \cdot \mathbf{T}_f, \quad (2.39)$$

josta voidaan tunnistaa ensimmäinen termi ortogonaaliseksi diffuusiovuoksi ja toinen termi epäortogonaalisuuden aiheuttamaksi lisätermiksi. Yleisesti tätä termiä kutsutaan ristidiffuusioksi. Yhteydessä (2.39) käytetty lineaarinen interpolaatio diskretisoi ortogonaalinen osuuden toisen kertaluvun tarkkuudella, mutta epäortogonaalinen osuus vaatii vielä erillisen \mathbf{T}_f -vektorin muodostamisen. (Moukalled et al. 2016, s. 241–242)

Pintavektorin \mathbf{S}_f jakaminen \mathbf{E}_f - ja \mathbf{T}_f -komponentteihin voidaan suorittaa kolmella eri tavalla. Ensimmäisenä vaihtoehtona on projisoida \mathbf{S}_f ortogonaalisesti yksikkövektorin \mathbf{e} suuntaiselle suoralle siten, että

$$\mathbf{E}_f = (\mathbf{e} \cdot \mathbf{S}_f)\mathbf{e} = (S_f \cos \theta)\mathbf{e}, \quad (2.40)$$

jossa θ on vektorien \mathbf{S}_f ja \mathbf{d}_{CF} välinen kulma. Toisena vaihtoehtona on muodostaa \mathbf{E}_f kertomalla pintavektorin \mathbf{S}_f suuruus yksikkövektorilla \mathbf{e} , eli

$$\mathbf{E}_f = S_f \mathbf{e}, \quad (2.41)$$

joka on siis riippumaton pintojen välisestä kulmasta. Kolmantena vaihtoehtona on käyttää ylirelaksoitua menetelmää, jonka mukaisesti vektori \mathbf{E}_f määräytyy yhteydestä

$$\mathbf{E}_f = \left(\frac{S_f}{\cos \theta} \right) \mathbf{e} = \left(\frac{S_f^2}{S_f \cos \theta} \right) \mathbf{e} = \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{e} \cdot \mathbf{S}_f} \mathbf{e}. \quad (2.42)$$

Yleisesti ensimmäinen menetelmä tunnetaan minimikorjauksen nimellä, toinen ortogonaalisena korjauksena ja viimeinen nimensä mukaisesti ylirelaksoituna korjauksena. (Moukalled 2016, s. 242–243) Ylirelaksoitua korjausmenetelmää on havainnollistettu kuvassa 2.2.

Erilaisten epäortogonaalisten korjausten toimivuutta voidaan tarkastella ratkaisemalla yksinkertainen Laplacen yhtälö $\nabla \cdot \nabla\phi = 0$, joka ei sisällä lähdetermejä. Muuttamalla neliskulmaisen laskentaverkon epäortogonaalisuutta välillä 10 – 65° voidaan nostaa esille muutamia havaintoja. Pienillä epäortogonaalisuuden arvoilla kaikki kolme menetelmää supenevat samaan lopputulokseen, mutta epäortogonaalisuuden nosto arvoon 30° aiheuttaa

jo merkittävää oskillaatiota minimikorjauksen menetelmää käytettäessä. Yli 40° epäortogonaalisuus aiheuttaa sen, että minimikorjausmenetelmä ei enää suppene lainkaan ja ortogonaalinen menetelmä oskilloi, mutta suppenee lopulta. Mikäli epäortogonaalisuus on 65° , vain ylirelaksoitu menetelmä suppenee. Ylirelaksoidun menetelmän etuna on myös se, että menetelmä suppenee monotonisesti oikeaan arvoon. (Jasak 1996, s. 138–143) Edellä mainituista syistä on selkeää, että käytännön sovelluksiin kannattaa valita yhtälön (2.42) mukainen epäortogonaalisuuden korjaus. Tätä menetelmää käytetään myös tässä työssä, koska myöhemmin osoittautuu, että laskentaverkkojen epäortogonaalisuus on luonnostaan varsin suuri.

Yhtälössä (2.39) esiintyvälle ristidiffuusiolle voidaan ylirelaksoidun menetelmän tapauksessa muodostaa yhtälö

$$(\nabla\phi)_f \cdot \mathbf{T}_f = (\nabla\phi)_f \cdot (\mathbf{S}_f - \mathbf{E}_f) = (\nabla\phi)_f \cdot \left(\mathbf{n} - \frac{1}{\cos\theta} \mathbf{e} \right) S_f, \quad (2.43)$$

jonka vaikutus häviää, jos epäortogonaalisuutta ei ole. Epäortogonaalisen lisätermin sisältävä gradientti $\nabla\phi_f$ voidaan arvioida, jos tunnetaan gradientti kontrollitulavuuden keskipisteessä. Kontrollitulavuuksien C ja F välisellä rajapinnalla tarvittava gradientti $\nabla\phi_f$ voidaan interpoloida yhteydestä

$$\nabla\phi_f = g_C \nabla\phi_C + g_F \nabla\phi_F, \quad (2.44)$$

jossa g_C ja g_F määräytyvät kontrollitulavuuksien perusteella. Vaikeutena diffuusio-ongelmassa on se, että miten ristidiffuusiotermissä (2.43) esiintyvä gradientti lasketaan, koska tätä ei saada muodostettua suoraan arvojen ϕ_C ja ϕ_F avulla, vaan gradientti $\nabla\phi_f$ täytyy muodostaa erikseen. (Moukalled et al. 2016, s. 244–245) Oletetaan gradientti $\nabla\phi_f$ tunnetuksi tässä vaiheessa, sillä tarkemmin gradienttien laskentaan perehdytään luvussa 2.4.

Diffuusioyhtälön (2.35) sisältämästä diffuusio kertoimesta Γ_f^ϕ ei ole kuitekaan vielä tehty päätelmiä, eikä ole oletettavaa, että diffuusio kerroin olisi vakio koko laskenta-alueessa tai edes yksittäisen kontrollitulavuuden välillä. Tästä syystä tarvitaan keino interpoloida diffuusio kerroin kontrollitulavuuksien välisille pinnoille.

Olettamalla diffuusio kertoimen muuttuvan lineaarisesti kontrollitulavuuden sisällä, voidaan johtaa yhteys

$$\Gamma_f^\phi = (1 - g_f) \Gamma_C^\phi + g_f \Gamma_F^\phi, \quad (2.45)$$

jossa esiintyvä interpolaatio kerroin g_f saadaan yhteydestä

$$g_f = \frac{d_{Cf}}{d_{Cf} + d_{fF}}. \quad (2.46)$$

Interpolaatiokertoimessa olevat vakiot määräytyvät kontrollitilavuuksien keskipisteiden välimatkan perusteella. (Moukalled et al. 2016, s. 224–225)

Diffuusioyhtälön (2.35) lopullinen diskretisointi on nyt mahdollista. Yhtälössä oleva summatermi voidaan kirjoittaa yhteyden (2.39) avulla muodossa

$$\sum_{f=1}^{n(f)} [-(\Gamma^\phi \nabla \phi)_f \cdot \mathbf{S}_f] = \sum_{f=1}^{n(f)} \left[-\Gamma_f^\phi E_f \frac{(\phi_F - \phi_C)}{d_{CF}} \right] + \sum_{f=1}^{n(f)} [-(\Gamma^\phi \nabla \phi)_f \cdot \mathbf{T}_f], \quad (2.47)$$

ja merkitsemällä lyhyesti geometrista diffuusiokerrointa yhteydellä $D_f = E_f/d_{CF}$, saadaan summatermi (2.47) lopulliseen muotoon

$$\sum_{f=1}^{n(f)} [-(\Gamma^\phi \nabla \phi)_f \cdot \mathbf{S}_f] = \sum_{f=1}^{n(f)} [\Gamma_f^\phi D_f (\phi_C - \phi_F)] + \sum_{f=1}^{n(f)} [-(\Gamma^\phi \nabla \phi)_f \cdot \mathbf{T}_f]. \quad (2.48)$$

Ottamalla huomioon alkuperäisen diffuusioyhtälön sisältämä lähdetermi, voidaan kirjoittaa edellä mainittujen yhteyksien avulla jokaiselle kontrollitilavuudelle yhtälö

$$a_C \phi_C + \sum_{F=1}^{n(F)} a_F \phi_F = b_C, \quad (2.49)$$

jossa $n(F)$ kuvaa kontrollitilavuuden naapureiden lukumäärää. Tarvittavat a_C - ja a_F -kerroimet saadaan geometrinen diffuusiokerroimien avulla, eli diagonaalikerroin yhteydestä

$$a_C = \sum_f^{n(f)} \Gamma_f^\phi D_f \quad (2.50)$$

ja naapurikerroimet yhteydestä

$$a_F = -\Gamma_f^\phi D_f. \quad (2.51)$$

Oikeanpuoleiseen vakiokerroimeen b_C voidaan lisätä lähdetermin ja myös epäortogonaalisuuden aiheuttama lisätermi, jolloin saadaan

$$b_C = Q_C^\phi V_C + \sum_{f=1}^{n(f)} [(\Gamma^\phi \nabla \phi)_f \cdot \mathbf{T}_f]. \quad (2.52)$$

Epäortogonaalisuus aiheuttaa kuitenkin sen lisävaivan, että yhtälöistä (2.49) saatava yhtälöryhmä on ratkaistava iteratiivisesti hyödyntämällä esimerkiksi alirelaksoinnin periaatetta. (Moukalled et al. 2016, 245–257) Tähän periaatteeseen palataan työssä hieman

myöhemmin. Tulokset (2.50–2.52) ovat yleisiä, ja näitä voidaan soveltaa tarpeen mukaan eri yhtälöille. Lähde-termien vaikutus otetaan kuitenkin huomioon vain kertaalleen.

Diffuusiotermin diskretisoinnissa nousi siis esille ongelma, että jos laskentaverkossa esiintyy epäortogonaalisuutta, joka on pikemminkin sääntö kuin poikkeus insinöörisovel-luksissa, ei välttämättä saavuteta toisen kertaluvun tarkkuutta. Edellä esitetty alirelaksoitu menetelmä on näillä näkymin paras tapa ottaa huomioon laskentaverkon epäortogonaali-suus. Tämän lisäksi epäortogonaalisuuden korjaamisessa tarvittu gradientti $\nabla\phi_f$ saattaa sisältää oman virhelähteensä, jos tätä ei määritetä toisen kertaluvun tarkkuudella.

2.4 Gradienttitermi

Edellisessä osiossa nousi esille tarve määrittää gradientti $\nabla\phi_f$ tarkasti. Jos halutaan yllä-pitää diskretisoinnin toisen kertaluvun tarkkuutta, tulee myös gradienttien olla toisen ker-taluvun tarkkuudella laskettuja. Tästä syystä tässä osiossa esitetään kaksi eri menetelmää gradienttien laskemiseen, ensin Greenin–Gaussin menetelmä ja myöhemmin pienimmän neliösumman menetelmä.

Greenin–Gaussin lauseen nojalla kontrollitilavuuden yli muodostettu tilavuusintegraali voidaan muuttaa pintaintegraaliksi, minkä seurauksena gradientin tilavuusintegraali voi-daan lausua muodossa

$$\int_V \nabla\phi dV = \oint_{\partial V} \phi d\mathbf{S}, \quad (2.53)$$

ja kontrollitilavuuden yli olevalle keskimääräiselle gradientille voidaan olettaa pätevän

$$\overline{\nabla\phi} V = \int_V \nabla\phi dV. \quad (2.54)$$

Kontrollitilavuuden keskimääräinen gradientti saadaan edellä mainittuja ominaisuuksia (2.53) ja (2.54) hyödyntäen pintaintegraalista

$$\overline{\nabla\phi}_C = \frac{1}{V_C} \oint_{\partial V} \phi_f \mathbf{S}_f, \quad (2.55)$$

ja lyhyesti merkitsemällä keskimääräistä gradienttia ilman yläviivaa saadaan se diskreet-tiin muotoon

$$\nabla\phi_C = \frac{1}{V_C} \sum_f^{n(f)} \phi_f \mathbf{S}_f, \quad (2.56)$$

mutta yhtälössä (2.56) esiintyvien kontrollitilavuuksien välisten rajapintojen pintakeskiöiden ϕ_f -arvot ovat kuitenkin vielä tuntemattomia. Tästä syystä gradientin laskeminen hieman hankaloituu. (Moukalled et al. 2016, s. 244–245) Greenin–Gaussin gradientin (2.56) tarkkuutta tarkastellaan myöhemmin tässä luvussa. Keskitytään nyt siihen, että miten gradientissa esiintyvät ϕ_f -arvot perinteisesti määritetään Greenin–Gaussin menetelmällä.

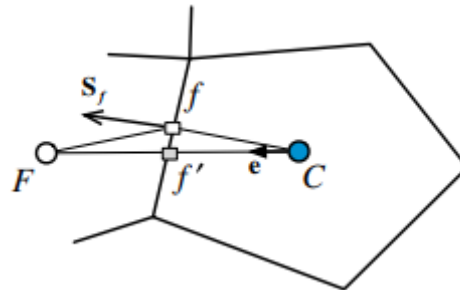
Mikäli mielenkiinnon kohteena olevan kontrollitilavuuden keskipisteen ϕ_C -arvo ja myös kontrollitilavuutta ympäröivien naapureiden ϕ_F -arvot tunnetaan, voidaan muuttujan ϕ_f -arvo interpoloida kontrollitilavuuksien väliselle rajapinnalle yhteydestä

$$\phi_f = g_C \phi_C + (1 - g_C) \phi_F, \quad (2.57)$$

jossa interpolaatiokerroimelle g_C saadaan arvo lausekkeesta

$$g_C = \frac{\|\mathbf{x}_F - \mathbf{x}_f\|}{\|\mathbf{x}_F - \mathbf{x}_C\|} = \frac{d_{Ff}}{d_{FC}} \quad (2.58)$$

tunnettujen etäisyyksien d_{Ff} ja d_{FC} avulla. Huomionarvoista on, että interpoloinnilla saavutetaan toisen kertaluvun tarkkuus vain siinä erikoistapauksessa, jos pintakeskiö ja keskipisteiden välille muodostettu suora kohtaavat, jota todellisuudessa on vaikea saavuttaa. (Moukalled et al. 2016, s. 275–276)



Kuva 2.3 Kontrollitilavuuksien välisen pinnan vinouma (Moukalled et al. 2016, s. 254)

Tähän asti johdetuissa yhtälöissä on oletettu, että kahden kontrollitilavuuden keskipisteen välinen vektori \mathbf{d}_{CF} kulkee näiden välisen pinnan pintakeskiön läpi. Jos näin ei käy, voidaan Moukalled et al. (2016, s. 254) mukaan tämä vinouma ottaa huomioon gradienttiin pohjautuvalla korjauksella

$$\phi_f = \phi_{f'} + (\nabla\phi)_{f'} \cdot \mathbf{d}_{f'f}, \quad (2.59)$$

jossa $\mathbf{d}_{f'f}$ kuvaa leikkauspisteen ja todellisen pintakeskiön välistä vektoria ja $(\nabla\phi)_{f'}$ leikkauspisteen gradienttia, joka täytyy vielä erikseen interpoloida leikkauspisteeseen f' . Vinouman aiheuttamaa lisähaastetta gradienttimenetelmille on havainnollistettu kuvassa 2.3.

Vinouman vuoksi gradientti ja sen laskemisessa tarvittavat rajapintojen ϕ_f -arvot on ratkaistava iteratiivisesti. Iteratiivisessa menettelyssä voidaan lähteä liikkeelle pinnan normaalin suuntaisen yksikkövektorin \mathbf{n} ja vektorin $\mathbf{d}_{f'f} = \mathbf{x}_f - \mathbf{x}_{f'}$, välisestä ortogonaalisuusehdosta

$$(\mathbf{x}_f - \mathbf{x}_{f'}) \cdot \mathbf{n} = 0. \quad (2.60)$$

Koska leikkauspiste f' on pisteiden C ja F välisellä suoralla, ja samalla yksikkövektorin \mathbf{e} suuntainen, voidaan hyödyntää tätä tietoa muodostamalla vektori leikkauspisteen f' ja keskipisteen C välille, minkä seurauksena

$$\mathbf{d}_{Cf'} = (\mathbf{x}_{f'} - \mathbf{x}_C) = n_k \mathbf{e}, \quad (2.61)$$

missä n_k on jokin kerroin. Yhdistämällä molemmat ehdot saadaan vektorille $\mathbf{x}_{f'}$ lauseke

$$\mathbf{x}_{f'} = \frac{\mathbf{x}_f \cdot \mathbf{n}}{\mathbf{e} \cdot \mathbf{n}} \mathbf{e}. \quad (2.62)$$

Rajapinnan ϕ_f -arvo sen todellisessa pintakeskiössä saadaan nyt yhtälöstä

$$\begin{aligned} \phi_f &= g_C [\phi_C + (\nabla\phi)_C \cdot (\mathbf{x}_f - \mathbf{x}_C)] + (1 - g_C) [\phi_F + (\nabla\phi)_F \cdot (\mathbf{x}_f - \mathbf{x}_F)] \\ &= \phi_{f'} + g_C (\nabla\phi)_C \cdot (\mathbf{x}_f - \mathbf{x}_C) + (1 - g_C) (\nabla\phi)_F \cdot (\mathbf{x}_f - \mathbf{x}_F), \end{aligned} \quad (2.63)$$

jossa kaksi viimeistä termiä ovat vinoudesta johtuvat korjaustermit. Korjauksessa tarvittava interpolointikerroin saadaan yhteydestä

$$g_C = \frac{\|\mathbf{x}_F - \mathbf{x}_{f'}\|}{\|\mathbf{x}_F - \mathbf{x}_C\|} = \frac{d_{Ff'}}{d_{FC}}. \quad (2.64)$$

Gradientti on edellä johdetun ansiosta mahdollista ratkaista iteratiivisesti aloittaen yhtälöstä (2.57) ja laskemalla tästä saaduilla arvoilla gradientti tuloksesta (2.56). Tähän voidaan tehdä lisäkorjaus yhtälön (2.63) avulla, mistä saatujen parannettujen arvojen perusteella voidaan taas määrittää uusi tarkempi gradientti tuloksesta (2.56). Menetelmään on kuitenkin myös toisia vaihtoehtoja. Esimerkiksi leikkauspiste f' voidaan pakottaa keskelle pisteiden C ja F välistä suoraa. Kolmantena mahdollisuutena on valita piste f' siten, että vektorin $\mathbf{x}_f - \mathbf{x}_{f'}$ pituus on mahdollisimman pieni. (Moukalled et al. 2016, s. 277–280) Kysymykseksi nousee, miten tarkka Greenin–Gaussin gradienttimenetelmä on todellisuudessa ja voidaanko toisen kertaluvun tarkkuus saavuttaa muissa kuin alkeistapauksissa.

Greenin–Gaussin menetelmän voidaan todistaa olevan kertalukua $\min(s, 1)$, jos laskentaverkon tihentämisen ansiosta sen vinouma vähenee nopeudella $O(h^s)$. Yleisesti säännöllisissä laskentaverkoissa vinouman ajatellaan vähenevän nopeudella $O(h)$, eli tällöin

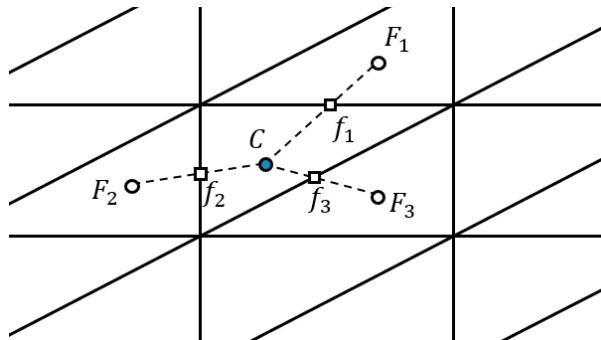
gradienttimenetelmä kykenee vain ensimmäisen kertaluvun tarkkuuteen. Greenin–Gaussin gradientin erikoispiirteenä on kuitenkin sen hyvä tarkkuus säännöllisissä laskentaverkoissa, koska Taylorin sarjakehitelmän avulla voidaan todistaa vastakkaisten pintojen aiheuttamien virheiden kumoavan toisensa, jolloin Greenin–Gaussin gradienttimenetelmän kertaluku on parhaimmillaan $O(h^2)$. Tätä ei voida yleistää esimerkiksi laskenta-alueen reunamille, koska näillä alueilla säännöllisessä laskentaverkossa esiintyy epäkeskeisyyttä ϑ , ja tämä voidaan määrittää yhteydestä

$$\vartheta = \frac{\|\mathbf{x}_{f'} - \mathbf{m}_f\|}{\|\mathbf{x}_F - \mathbf{x}_C\|}, \quad (2.65)$$

jossa \mathbf{m}_f kuvaa kontrollitilavuuksien keskipisteiden avulla saatua vektoria

$$\mathbf{m}_f = \frac{1}{2}(\mathbf{x}_C + \mathbf{x}_F). \quad (2.66)$$

Greenin–Gaussin menetelmän tarkkuus voidaan havaita kärsivän reunamilla, koska näillä esiintyy kaavan (2.65) mukaista epäkeskeisyyttä. Reunoilla laskentasolmu sijoitetaan reunapinnan keskelle, eikä tilavuuden keskelle, kuten laskenta-alueen sisällä voidaan tehdä. Seinämällä ei voida myöskään käyttää toisen kertaluvun tarkkuudella interpoloitua arvoa, jolloin voidaan todistaa menetelmän virheen pienenevän nopeudella $O(h)$. Jos kahden kontrollitilavuuden välisen pinnan leikkauspisteestä on sama matka molempien kontrollitilavuuksien keskelle, eli yhtälön (2.65) mukainen epäkeskeisyys on nolla, toisen kertaluvun tarkkuus palautuu. (Syrakos et al. 2017, s. 3–8)



Kuva 2.4 Säännöllisistä nelikulmioista nelikulmioitu laskentaverkko, jossa Greenin–Gaussin gradientti menettää tarkkuuttaan. Mukailtu lähteestä (Syrakos et al. 2017, s. 8).

Toinen ongelma voidaan saada esille, jos esimerkiksi 2D-tapauksessa säännölliset nelikulmiot pilkotaan diagonaalisesti kahdeksi kolmioksi, ja vaikka lopputuloksena on ilman vinoumaa tai epäkeskeisyyttä oleva laskentaverkko, odotettu toisen kertaluvun tarkkuus menetetään ja menetelmä on vain ensimmäistä kertalukua. Epäsäännöllisen laskentaverkon tapauksessa tilanne on vieläkin haastavampi, sillä vinouskorjaamattoman gradientin virhe ei suppene ja tilanne paranee vasta, jos suoritetaan n kappaletta korjausiteraatioita, kunnes gradientti suppenee oikeaan arvoonsa. Tällöin menetelmän tarkkuus palautuu tasolle $O(h)$. Taustalla on ajatus, että täysin epäsäännöllisten laskentaverkkojen vinouma

pienenee, mutta ei välttämättä suppene mihinkään tiettyyn arvoon, koska luodut laskentaverkot ovat similaarisia riippumatta niiden tiheydestä. (Syrakos et al. 2017, s. 8–10) Edellä mainittua tapausta, jossa Greenin–Gaussin gradienttimenetelmän toimivuus kärsii, on havainnollistettu kuvassa 2.4.

Greenin–Gaussin gradientin sijaan on mahdollista ratkaista gradientti $\nabla\phi_C$ suoraan pienimmän neliösumman menetelmällä olettamalla kontrollitulavuuden sisälle lineaarinen variaatio

$$\phi_F = \phi_C + (\nabla\phi)_C \cdot (\mathbf{x}_F - \mathbf{x}_C). \quad (2.67)$$

Esille nousee kuitenkin ongelma, koska yhteydestä (2.67) saatava yhtälöiden määrä on pienempi kuin jokaista kontrollitulavuutta ympäröivien naapureiden lukumäärä. Tästä syystä gradientin $\nabla\phi_C$ ratkaisemista varten on muodostettava kohdefunktio

$$G_C = \sum_{k=1}^{n(k)} \left\{ w_k [\phi_{F_k} - (\phi_C + \nabla\phi_C \cdot \mathbf{d}_{CF_k})]^2 \right\}, \quad (2.69)$$

ja avaamalla sen termit saadaan minimoitavaksi funktioksi

$$G_C = \sum_{k=1}^{n(k)} \left\{ w_k \left[\Delta\phi_k - \left(\Delta x_k \left(\frac{\partial\phi}{\partial x} \right)_C + \Delta y_k \left(\frac{\partial\phi}{\partial y} \right)_C + \Delta z_k \left(\frac{\partial\phi}{\partial z} \right)_C \right) \right]^2 \right\}. \quad (2.70)$$

Funktiossa (2.70) käytetään yhteyksiä

$$\Delta\phi_k = \phi_{F_k} - \phi_C$$

$$\Delta x_k = \mathbf{d}_{CF_k} \cdot \mathbf{i}$$

$$\Delta y_k = \mathbf{d}_{CF_k} \cdot \mathbf{j}$$

$$\Delta z_k = \mathbf{d}_{CF_k} \cdot \mathbf{k}, \quad (2.71)$$

ja w_k on painokerroin, jonka avulla voidaan hienosäätää, miten paljon kontrollitulavuuden yksittäisillä lähinaapureilla on vaikutusta gradientin arvoon. (Moukalled et al. 2016, s. 285–286)

Kohdefunktiossa (2.70) muuttujina ovat gradientin komponentit $\partial\phi/\partial x$, $\partial\phi/\partial y$ ja $\partial\phi/\partial z$, joten minimin olemassaolon välttämätön edellytys on

$$\frac{\partial G_C}{\partial \left(\frac{\partial\phi}{\partial x} \right)} = \frac{\partial G_C}{\partial \left(\frac{\partial\phi}{\partial y} \right)} = \frac{\partial G_C}{\partial \left(\frac{\partial\phi}{\partial z} \right)} = 0, \quad (2.72)$$

ja minimiehdosta (2.72) saadaan nyt kolme ratkaistavaa yhtälöä derivoimalla funktiota G_c , minkä seurauksena saadaan yhtälöryhmä

$$\begin{aligned} \sum_{k=1}^{n(k)} \left\{ 2w_k \Delta x_k \left[-\Delta \phi_k + \Delta x_k \left(\frac{\partial \phi}{\partial x} \right)_c + \Delta y_k \left(\frac{\partial \phi}{\partial y} \right)_c \Delta z_k \left(\frac{\partial \phi}{\partial z} \right)_c \right] \right\} &= 0 \\ \sum_{k=1}^{n(k)} \left\{ 2w_k \Delta y_k \left[-\Delta \phi_k + \Delta x_k \left(\frac{\partial \phi}{\partial x} \right)_c + \Delta y_k \left(\frac{\partial \phi}{\partial y} \right)_c \Delta z_k \left(\frac{\partial \phi}{\partial z} \right)_c \right] \right\} &= 0 \\ \sum_{k=1}^{n(k)} \left\{ 2w_k \Delta z_k \left[-\Delta \phi_k + \Delta x_k \left(\frac{\partial \phi}{\partial x} \right)_c + \Delta y_k \left(\frac{\partial \phi}{\partial y} \right)_c \Delta z_k \left(\frac{\partial \phi}{\partial z} \right)_c \right] \right\} &= 0. \end{aligned} \quad (2.73)$$

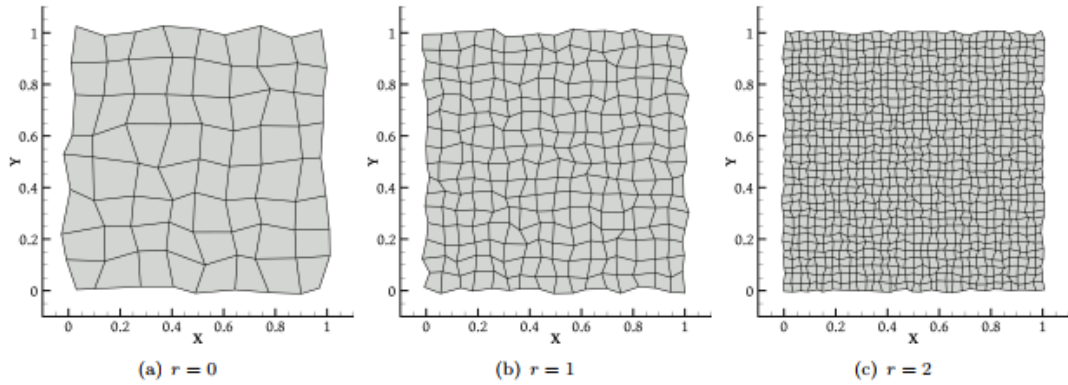
Yhtälöt (2.73) voidaan kasata helpommin käsiteltävään matriisimuotoon

$$\begin{bmatrix} \sum_{k=1}^{n(k)} w_k \Delta x_k \Delta x_k & \sum_{k=1}^{n(k)} w_k \Delta x_k \Delta y_k & \sum_{k=1}^{n(k)} w_k \Delta x_k \Delta z_k \\ \sum_{k=1}^{n(k)} w_k \Delta y_k \Delta x_k & \sum_{k=1}^{n(k)} w_k \Delta y_k \Delta y_k & \sum_{k=1}^{n(k)} w_k \Delta y_k \Delta z_k \\ \sum_{k=1}^{n(k)} w_k \Delta z_k \Delta x_k & \sum_{k=1}^{n(k)} w_k \Delta z_k \Delta y_k & \sum_{k=1}^{n(k)} w_k \Delta z_k \Delta z_k \end{bmatrix} \begin{bmatrix} \left(\frac{\partial \phi}{\partial x} \right)_c \\ \left(\frac{\partial \phi}{\partial y} \right)_c \\ \left(\frac{\partial \phi}{\partial z} \right)_c \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{n(k)} w_k \Delta x_k \Delta \phi_k \\ \sum_{k=1}^{n(k)} w_k \Delta y_k \Delta \phi_k \\ \sum_{k=1}^{n(k)} w_k \Delta z_k \Delta \phi_k \end{bmatrix}, \quad (2.74)$$

jolle saadaan ratkaisu lineaarialgebran keinoin, esimerkiksi kääntämällä vasemmanpuoleinen kerroinmatriisi. (Moukalled et al. 2016, s. 286–287) Pienimmän neliösumman menetelmä on selkeästi monimutkaisempi tapa ratkaista gradientti, mutta menetelmä on ehdottomasti käyttämisen arvoinen kuten seuraavaksi ilmenee.

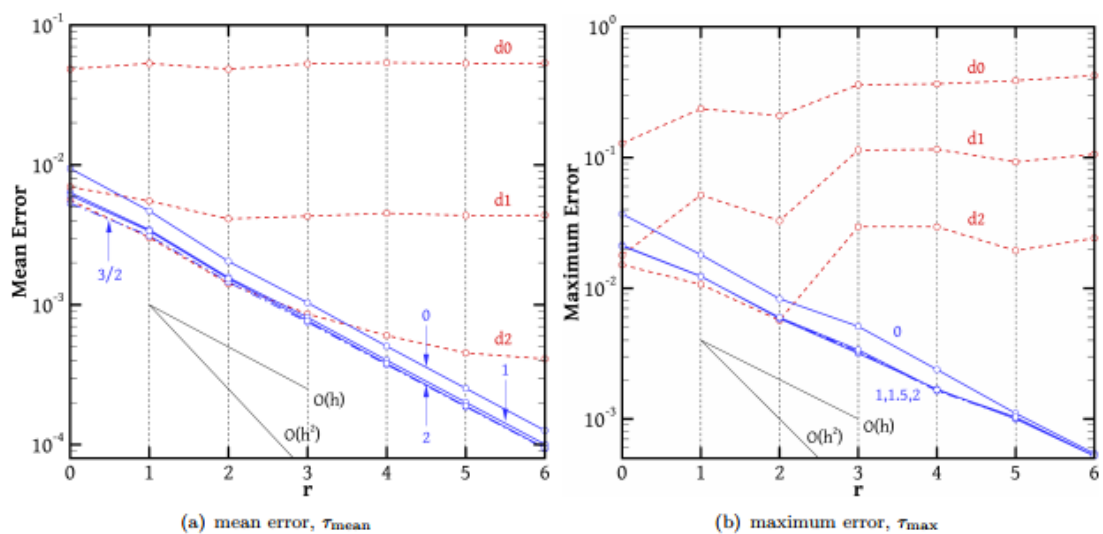
Pienimmän neliösumman menetelmällä saatu gradientti on merkittävästi Greenin–Gaussin menetelmällä saatua gradienttia tarkempi esimerkiksi epäsäännöllisen laskentaverkon tapauksessa. Pienimmän neliösumman menetelmän tarkkuuden voidaan todistaa olevaan vähintään $O(h)$ kaikille laskentaverkkotyypeille, myös epäsäännöllisille mielivaltaisille 3D-monikulmioille. Molempien menetelmien toimivuutta voidaan tarkastella laskemalla yksinkertainen malliongelma $\phi = \tanh(x) \tanh(y)$ kuvan 2.5 mukaisissa laskentaverkoissa, joissa laskentatilavuuksien (tässä tapauksessa pintojen) muodot ovat täysin epäsäännöllisiä. Osoittautuu, että pienimmän neliösumman menetelmä kykenee laskemaan gradientin siten, että virheen pienenemisnopeus on $O(h)$ sekä maksimivirheen että keskivirheen osalta. Sen sijaan Greenin–Gaussin menetelmän virhe ei pahimmillaan suppene lainkaan. Ilmiö voidaan selittää sillä, epäsäännöllisten laskentaverkkojen vinouma tai epäkeskeisyys ei merkittävästi pienene laskentaverkkoa tihennettäessä. (Syrakos et al.

2017, s. 25–27) Kuvassa 2.6 on esitetty edellä mainittu menetelmien eroavaisuus, missä siniset viivat vastaavat eri painokertoimilla w_k saatuja tuloksia. Lisäksi punaisella näkyvä d_0 vastaa korjaamatonta Greenin–Gaussin menetelmää, d_1 yhden korjausiteraation menetelmää ja d_2 vastaavasti kahden korjausiteraation menetelmää.



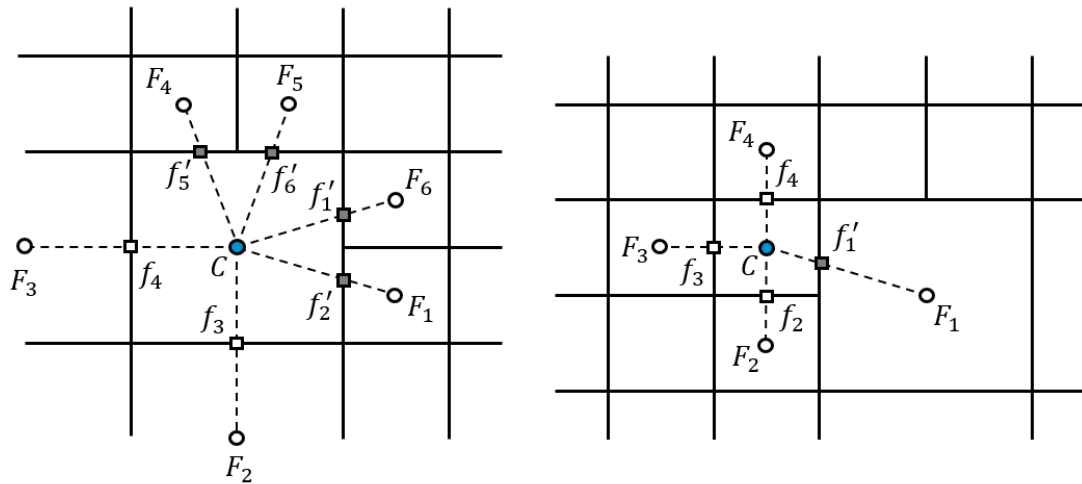
Kuva 2.5 Gradienttimenetelmien tutkimisessa käytettyjä laskentaverkkoja (Syrakos et al. 2017, s. 26)

Vaikka keskivirhe pieneneekin varsin hyvin käytettäessä kahta korjausiteraatiota Greenin–Gaussin menetelmässä, Syrakos et al. (2017, s. 30) huomauttaa, että esimerkiksi yksinkertaisen Poissonin yhtälön ratkaisemisessa pienimmän neliösumman menetelmä on laskennallisesti halvempaa kuin vain yhtä korjausiteraatiota käyttävä Greenin–Gaussin menetelmä. Tästä syystä voidaan pitää järkevänä pienimmän neliösumman menetelmän käyttöä ja tätä sovelletaankin myöhemmin tässä työssä. Alun perin ilman painokertoimia olevan menetelmän on esittänyt Muzafferija & Gosman (1997, s. 769–771) adaptiivisen kontrollitulavuusmenetelmän yhteydessä, joten menetelmä vaikuttaa varsin lupaavalta myös tämän työn kannalta.



Kuva 2.6 Pienimmän neliösumman menetelmän ja Greenin–Gaussin menetelmän keskimääräisen virheen ja maksimivirheen käyttäytyminen (Syrakos et al. 2017, s. 27)

Pienimmän neliösumman menetelmä ei kuitenkaan ole täysin ongelmaton, vaikka menetelmän edut ovat selvät. Syrakos et al. (2017, s. 21–25) nostaa esille kuvan 2.7 mukaisen ongelmatapauksen, jossa kontrollitilavuus toimii erikokoisten kontrollitilavuuksien välisenä rajapintana, mikä johtaa siihen, että painokertoimien vuoksi esimerkiksi kuvan 2.7 vasemmanpuoleisessa tapauksessa gradienttiin painotetaan enemmän pienempien laskentatilavuuksien vaikutusta kuin suurempien. Tästä syystä painokertoimiin kannattaa paineutua pikaisesti.



Kuva 2.7 Rajapintakontrollitilavuuksien aiheuttama haaste pienimmän neliösumman menetelmälle. Mukailtu lähteestä (Syrakos et al. 2017, s. 23).

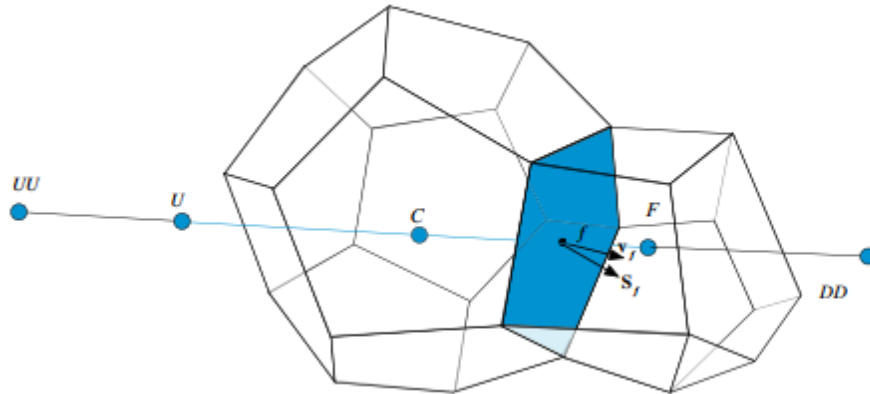
Tyypillisesti pienimmän neliösumman menetelmässä käytetään painokertoimina kontrollitilavuuksien välisen etäisyyden käänteisarvoa $\|\mathbf{x}_F - \mathbf{x}_C\|$ korotettuna johonkin potenssiin q , eli

$$w_k = \frac{1}{\|\mathbf{x}_F - \mathbf{x}_C\|^q}. \quad (2.75)$$

Tästä johtuen lähempänä kontrollitilavuuden keskipistettä olevien naapureiden vaikutus lisääntyy, mikä ei aina ole hyödyllistä. Asettamalla $q = 1$ menetelmä luonnostaan painottaa kaikkia tasapuolisesti. Sen sijaan valitsemalla $q = 2$ saadaan suurempi vaikutus lähellä olevista kontrollitilavuuksista. Mikäli menetelmää halutaan käyttää ilman painofunktioita, valitaan $q = 0$, minkä seurauksena menetelmä painottaa kauempana olevia kontrollitilavuuksia enemmän kuin lähellä olevia. Erikoistapauksessa $q = 3/2$ voidaan saada tarkkuudeltaan toisen kertaluvun menetelmä, jos kontrollitilavuuden naapurit ovat aina pareittain vastakkaisilla puolilla kontrollitilavuutta. Näin voi käydä esimerkiksi säännöllisten laskentaverkkojen tapauksissa. Tätä ominaisuutta ei voi kuitenkaan yleistää kaikille laskentaverkkotyypeille. (Syrakos et al. 2017, s. 14–15)

2.5 Konvektiotermi

Tähän mennessä on saatu muodostettua sekä diffuusiotermeille että gradienteille toisen kertaluvun menetelmät, joiden tarkkuus on mahdollisimman riippumaton laskentaverkon muodosta. Seuraavaksi keskitytäänkin skalaarimuuttujan differentiaaliyhtälön konvektiotermin diskretisointiin tarkastelemalla yksinkertaistettua konvektio-ongelmaa.



Kuva 2.8 Mielivaltaisten kontrollitilavuuksien välinen rajapinta f (Moukalled et al. 2016, s. 457)

Tarkastellaan yhtälöstä (2.1) muodostettua stationääristä konvektiivista ongelmaa

$$\nabla \cdot (\rho \mathbf{u} \phi) = Q^\phi, \quad (2.76)$$

jonka diskretisoiduksi muodoksi voidaan johtaa

$$\sum_{f=1}^{n(f)} (\rho \mathbf{u} \phi)_f \cdot \mathbf{S}_f = Q_c^\phi V_C. \quad (2.77)$$

Yhtälössä vasemmalla puolella esiintyvä termi $(\rho \mathbf{u})_f \cdot \mathbf{S}_f$ voidaan tunnistaa itse asiassa kahden kontrollitilavuuden välisen rajapinnan f läpi meneväksi massavirraksi, eli

$$\dot{m}_f = (\rho \mathbf{u})_f \cdot \mathbf{S}_f. \quad (2.78)$$

Sijoittamalla diskretisoituun yhtälöön massavirran lauseke saadaan yksinkertainen muoto

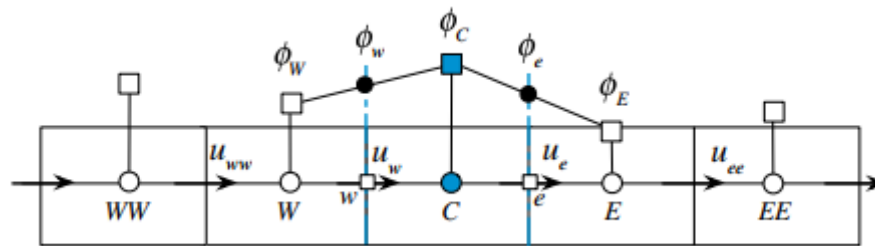
$$\sum_{f=1}^{n(f)} \dot{m}_f \phi_f = Q_c^\phi V_C, \quad (2.79)$$

jossa konvektiotermi muodostetaan massavirtojen \dot{m}_f ja rajapintojen arvojen ϕ_f avulla. Lopulta konvektio-ongelmasta saadaan jokaista kontrollitilavuutta kohden yhtälö

$$a_C \phi_C + \sum_F^{n(F)} a_F \phi_F = b_C. \quad (2.80)$$

Yhtälössä (2.80) esiintyviä kertoimia a_C ja a_F ei voida suoraan lausua arvojen ϕ_C ja ϕ_F perusteella kuten diffuusiotermin tapauksessa, vaan kertoimet ovat konvektiomenetelmästä riippuvia. Eräänä vaihtoehtona on muodostaa kertoimet alavirran painokertoimien menetelmällä, eli DWF-menetelmällä (engl. Downwind Weighing Factor Method). (Moukalled et al. 2016, s. 456–463) Kontrollitilavuuksien välistä rajapintaa on havainnollistettu kuvassa 2.8.

Edellä mainitusta syystä tässä työssä ei sen enempää keskitytä kertoimien a_C ja a_F määrittämiseen konvektiotapaukselle, vaan riittää, että ymmärretään konvektiotermin diskretisoinnin johtavan kyseisiin kertoimiin, jotka voidaan lisätä diffuusiotermeistä saataviin kertoimiin. Luonnollisesti lähdetermiä ei diskretisoida moneen kertaan, jos tämä on jo aiemmin suoritettu. Seuraavaksi tutustutaan erilaisiin konvektiomenetelmiin, joilla tarvittavia rajapintojen ϕ_f -arvoja voidaan määrittää.



Kuva 2.9 1D-laskentaverkko ja konvektiotermin lineaarinen variaatio kontrollitilavuuksien välillä (Moukalled et al. 2016, s. 370)

Yksinkertainen tapa tutkia yleisempien konvektiotermin diskretisointimenetelmien toimivuutta on ratkaista konvektio- ja diffuusiotermit sisältävä yksidimensionaalinen yhtälö

$$\frac{d(\rho u \phi)}{dx} - \frac{d}{dx} \left(\Gamma \phi \frac{d\phi}{dx} \right) = 0, \quad (2.81)$$

jolle tunnetaan analyttisenä ratkaisuna

$$\frac{\phi - \phi_W}{\phi_E - \phi_W} = \frac{e^{Pe_L \frac{x-x_W}{L}} - 1}{e^{Pe_L} - 1}. \quad (2.82)$$

Analyttisessä ratkaisussa ilmenevä dimensioton Pécletin luku saadaan yhteydestä

$$Pe_L = \frac{\rho u L}{\Gamma \phi}, \quad (2.83)$$

jossa esiintyvä $L = x_E - x_W$ kuvaa itänaapurin ja länsinaapurin välimatkaa. (Moukalled et al. 2016 s. 366–367) Koska analyttinen ratkaisu tunnetaan, voidaan verrata numeerista ratkaisua ja analyttistä ratkaisua keskenään ja saada ainakin jonkinasteinen ymmärrys konvektiomenetelmien toimintaan, joilla kontrollitilavuuksien välisten rajapintojen ϕ_f -arvoja pyritään ratkaisemaan. 1D-malliongelmaa on havainnollistettu kuvassa 2.9.

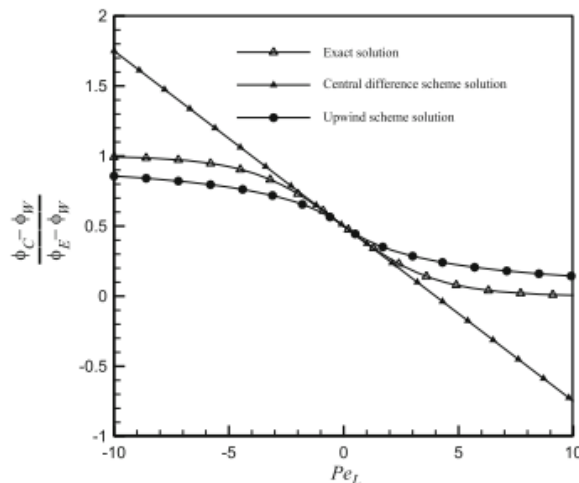
Yhtälö (2.81) voidaan ratkaista numeerisesti integroimalla se kontrollitilavuuden yli ja olettamalla muuttujalle ϕ lineaarinen profiili kontrollitilavuuksien välille kuvan 2.9 esittämällä tavalla. Lineaarisen profiilin avulla rajapinnan arvo saadaan yhteydestä

$$\phi_e = \phi_c + \frac{(\phi_E - \phi_C)}{(x_E - x_C)}(x_e - x_C), \quad (2.84)$$

joka yleisesti tunnetaan Taylorin sarjakehitelmien perusteella toisen kertaluvun interpolointimenetelmäksi. Tämän avulla voidaan johtaa ratkaisu

$$\frac{\phi_C - \phi_W}{\phi_E - \phi_W} = \frac{e^{\frac{Pe_L}{2}} - 1}{e^{Pe_L} - 1}, \quad (2.85)$$

mutta valitettavasti lineaarisen profiilin käyttäminen aiheuttaa kuvan 2.10 mukaisen ongelman. Numeerinen ratkaisu vastaa tarkkaa ratkaisua vain hyvin pienillä Pécletin luvuilla ja kasvaa rajatta, eikä siten rajoitu välille $[0,1]$, kuten analyttinen ratkaisu. Tästä syystä lineaarisen interpolointimenetelmän voidaan sanoa tuottavan epäfysikaalisia tuloksia. (Moukalled et al. 2016 s. 368–373) Vaikka kyseessä onkin toisen kertaluvun menetelmä, nousee kysymys, voidaanko löytää parempi vaihtoehto, joka estää edellä mainitun rajapintojen epäfysikaalisuuden ja säilyttää toisen kertaluvun tarkkuuden.



Kuva 2.10 Keskeisdifferenssimenetelmän, ylävirran menetelmän ja analyttisen ratkaisun vertailu (Moukalled et al. 2016, s. 378)

Toisena vaihtoehtona lineaariselle interpoloinnille on olettaa kuvan 2.11 mukaisesti askelittain muuttuva, pisteestä C katsoen aina ylävirrasta (engl. upwind) haettu profiili

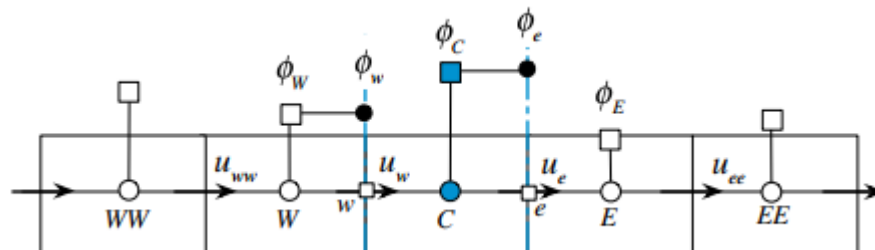
$$\phi_e = \begin{cases} \phi_C, & \dot{m}_e > 0 \\ \phi_E, & \dot{m}_e < 0 \end{cases}, \quad \phi_w = \begin{cases} \phi_C, & \dot{m}_w > 0 \\ \phi_W, & \dot{m}_w < 0 \end{cases} \quad (2.86)$$

missä \dot{m}_e ja \dot{m}_w ovat rajapintojen läpi meneviä massavirtoja. Ylävirran suunta voidaan päätellä massavirran etumerkistä. Profiilien (2.86) käyttäminen johtaa numeeriseen ratkaisuun

$$\frac{\phi_C - \phi_W}{\phi_E - \phi_W} = \frac{2 + \max(-Pe_L, 0)}{4 + \max(-Pe_L, 0) + \max(Pe_L, 0)}, \quad (2.87)$$

jonka voidaan havaita aina rajoittuvan välille $[0,1]$, kuten kuva 2.10 osoittaa. Menetelmän tarkkuus on kuitenkin heikko, kun tätä verrataan analyttiseen ratkaisuun, koska numeerinen ratkaisu vastaa analyttistä tulosta vain erittäin pienillä Pécletin luvuilla. (Moukalled et al. 2016 s. 376–378)

Lineaarisen interpoloinnin eli keskeisdifferenssimenetelmän heikkoa suorituskykyä voidaan selittää sillä, että konvektio on luonnostaan hyvin suunnasta riippuvaa. Olettamalla lineaarinen variaatio pisteiden C ja E välille tehdään implisiittinen oletus, että molemmilla pisteillä on yhtä suuri vaikutus näiden välisen rajapinnan arvoon, kuten diffuusion tapauksessa voitiin olettaa. Todellisuudessa suuntariippuvuuden takia ylävirran arvolla on suurempi vaikutus kuin alavirran arvolla. Tästä syystä askelmainen ylävirran profiili toimii paremmin, kun Pécletin luku kasvaa. Voidaan todistaa, että keskeisdifferenssimenetelmä johtaa epäfysikaalisiin tuloksiin, kun $Pe > 2$. (Moukalled et al. 2016, s. 373–375)



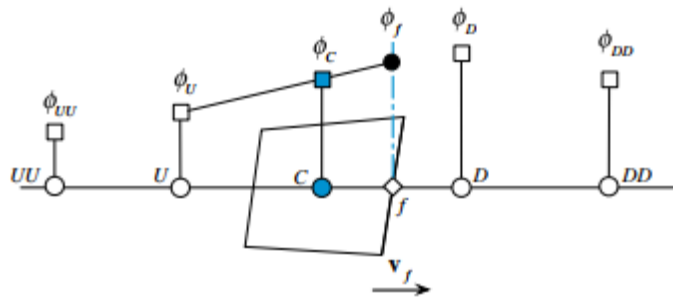
Kuva 2.11 Ylävirran menetelmä (Moukalled et al. 2016, s. 376)

Taylorin sarjakehitelmien avulla on mahdollista saada selville keskeisdifferenssimenetelmän ja ylävirran menetelmän diskretisointivirhe malliongelman (2.81) tapauksessa. Osoittautuu, että keskeisdifferenssimenetelmän virhe pienenee nopeudella $O(h^2)$ mutta ylävirran menetelmä vain $O(h)$, minkä seurauksena ylävirran menetelmä aiheuttaa ratkaisuun numeerista diffuusiota. Stabiiliustarkastelun jälkeen kuitenkin huomataan, että ylävirran menetelmä on aina stabiili, eivätkä numeeriset virheet voi kasvaa laskennan aikana rajatta. Keskeisdifferenssimenetelmälle tätä ominaisuutta ei voida taata edes yksinkertaisessa 1D-tapauksessa, jos Pécletin luku on suuri. (Moukalled et al. 2016, s. 380–388)

Kolmantena vaihtoehtona on hyödyntää lineaarista ekstrapolointia, jossa painotetaan ylävirran arvoja kuvan 2.12 mukaisesti. Profiilina voidaan tällöin käyttää

$$\phi_f = \phi_C + \frac{(\phi_C - \phi_U)}{(x_C - x_U)}(x_f - x_C), \quad (2.88)$$

jossa U kuvaa ylävirran kontrollitilavuuden keskipistettä ja C normaalisti mielenkiinnon kohteena olevan kontrollitilavuuden keskipistettä. Menetelmää kutsutaan lineaariseksi ylävirran menetelmäksi, jolla saavutetaan toisen kertaluvun tarkkuus, eli menetelmän virhe pienenee nopeudella $O(h^2)$. Menetelmä on myös stabiili malliongelman tapauksessa. (Moukalled et al. 2016, s. 388–392)



Kuva 2.12 Lineaarinen ylävirran menetelmä (Moukalled et al. 2016, s. 390)

Lukijalle voi tässä vaiheessa tulla kysymys, eikö lineaarisen ekstrapoloinnin sijaan voisi sovittaa toisen tai kolmannen asteen käyriä esimerkiksi pisteiden U , C ja D välille, jolloin virhe pienenee ainakin nopeudella $O(h^3)$. Näin voidaankin tehdä, ja toisen asteen käyrää hyödyntävä menetelmä yleisesti tunnetaan QUICK-menetelmänä (engl. Quadratic Upstream Interpolation for Convective Kinematics), mutta Moukalled et al. (2016, s. 392–395) huomauttaa QUICK-menetelmän olevan stabiili vain hyvin yksinkertaisissa 1D-tapauksissa.

Yleisesti Swebyn (1984, s. 997-998) mukaan kaikki ensimmäistä kertalukua tarkemmat konvektiomenetelmät aiheuttavat tulosta vääristelevää oskillaatiota numeeriseen ratkaisuun siinä esiintyvissä epäjatkuvuuskohdissa, ellei menetelmän voida todistaa olevan sellainen menetelmä, joka takaa ratkaisun totaalivariaation

$$TV(\phi^{t+\Delta t}) = \sum_k \|\phi_{k+1}^{t+\Delta t} - \phi_k^{t+\Delta t}\| \quad (2.89)$$

pienenevän aika-askeleiden välillä, eli toisin sanoen totaalivariaatioehto

$$TV(\phi^{t+\Delta t}) \leq TV(\phi^t) \quad (2.90)$$

takaa ratkaisun suppenevan fysikaalisesti oikeaan ratkaisuun. Tämän seurauksena konvektiotermin diskretisointi vaatii täysin erilaista näkökulmaa, eikä vain keinoa, jolla interpoloidaan tai ekstrapoloidaan rajapintojen pintakeskiöiden ϕ_f -arvot.

TVD-ehdon (engl. Total Variation Diminishing) toteuttavien menetelmien ideana on johdattaa ylemmän kertaluvun konvektiomenetelmästä sellainen muoto, jossa fysikaalisuus voidaan taata hyödyntämällä vuorajoittimien (engl. flux limiter) periaatetta. Vuorajoittimelle käytetään funktiota $\psi_k = \psi(r_k)$, jossa esiintyvä rajoitinparametri r_k saadaan lineaarisen skalaariyhtälön tapauksessa kahden peräkkäisen gradientin suhteesta, eli

$$r_k = \frac{\nabla\phi_{k-1/2}}{\nabla\phi_{k+1/2}}. \quad (2.91)$$

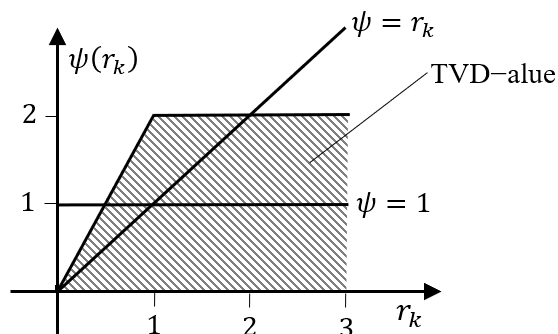
Pitkällisen yhtälöiden johtamisen jälkeen saadaan vuorajoittimelle TVD-ehto

$$0 \leq \left(\frac{\psi(r_k)}{r_k}, \psi(r_k) \right) \leq 2, \quad (2.92)$$

mikä graafisesti esitettynä on kuvan 2.13 mukainen alue. Tämän seurauksena vuorajoittimen $\psi(r_k)$ voidaan taata olevan TVD-alueella, kun se täyttää ehdot

$$\psi(r_k) = \begin{cases} 0, & r_k < 0, \\ \min(2r_k, 2), & r_k > 0, \end{cases} \quad (2.93)$$

minkä perusteella menetelmä redusoituu ensimmäisen kertaluvun menetelmäksi minimi- ja maksimipisteiden läheisyydessä, kun gradientti vaihtaa merkkiä, jolloin $r_k < 0$. (Sweby 1984, s. 999–1000) Kysymykseksi nousee, voidaanko koko TVD-alue käyttää hyödyksi ja miten tämä ylipäätään tapahtuu.



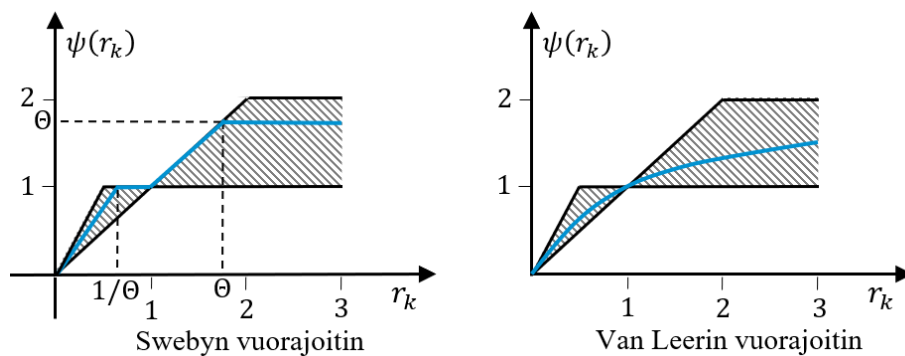
Kuva 2.13 Vuorajoittimien TVD-alue. Mukailtu lähteestä Sweby (1984, s. 1001).

Lähtökohtaisesti vuorajoitin kannattaa formuloida siten, että menetelmä kykenee pääasiassa toisen kertaluvun tarkkuuteen ja täyttää samalla myös TVD-ehdon. Haluttu toisen kertaluvun alue saadaan, kun vuorajoitin muodostetaan vuorajoittimien $\psi(r_k) = 1$ ja $\psi(r_k) = r_k$ painotettuna keskiarvona. Vuorajoittimen tulee kuitenkin aina kulkea pisteen (1,1) kautta. (Sweby 1984, s. 1000–1001) Itse asiassa Moukalled et al. (2016, s. 447–448) mukaan $\psi(r_k) = 1$ rajoitin kuvastaa keskeisdifferenssimenetelmää ja $\psi(r_k) = r_k$ lineaarista ylävirran menetelmää. Kyseiset rajoittimet on esitetty kuvassa 2.13.

Seuraavana ongelmana on selvittää, miten vuorajoitin muodostetaan rajoittamaan rajapintojen välisiä arvoja. Sweby (1984, s. 1002) ehdottaa aiemmin mainittujen ominaisuuksien perusteella toisen kertaluvun TVD-alueella paloittain jatkuvaa vuorajoitinta

$$\psi(r_k) = \max[0, \min(\Theta r_k, 1), \min(r_k, \Theta)], \quad (2.94)$$

jossa Θ -parametri voidaan valita vapaasti väliltä $1 \leq \Theta \leq 2$. Kuvassa 2.14 on esitetty kyseinen toisen kertaluvun TVD-alue, Swebyn rajoitinfunktio ja myös toinen suosittu Van Leerin rajoitinfunktio. Pelkkä vuorajoittimen formulointi ei vielä vie lähemmäksi rajapintojen arvojen määrittämistä, vaan vuorajoittimella saatiin varmistettua konvektiotermin fysikaalisuus.



Kuva 2.14 Swebyn ja Van Leerin ehdottamat rajoitinfunktiot. Mukailtu lähteestä (Sweby 1984, s. 1002).

Kontrollitulavuuksien välisten rajapintojen arvot voidaan TVD-viitekehyksessä määrittää ylävirran menetelmän ja tähän lisättävän vuomaisen termin yhdistelmänä, eli yhteydestä

$$\phi_f = \phi_C + \frac{1}{2} \psi(r_k) (\phi_D - \phi_C), \quad (2.95)$$

jossa kahden peräkkäisen gradientin suhde on esimerkiksi 1D-tapauksessa

$$r_k = \frac{\phi_C - \phi_U}{\phi_D - \phi_C}. \quad (2.96)$$

Mikäli vuorajoitin asetetaan TVD-alueen sisälle, konvektiomenetelmän voidaan tällöin luottaa tuottavan fysikaalisia tuloksia. Erikoistapauksissa $\psi(r_k) = 0$ saadaan palautettua perinteinen ylävirran menetelmä, $\psi(r_k) = r_k$ lineaarinen ylävirran menetelmä ja keskeisdifferenssimenetelmä, kun valitaan $\psi(r_k) = 1$. Ongelmana perinteisissä menetelmissä on se, että vain ylävirran menetelmä rajoittuu TVD-alueelle, sen sijaan muut menetelmät ovat osittain TVD-alueen ulkopuolella. (Moukalled et al. 2016, s. 444–448)

Erilaisia rajoitinfunktioita on lukuisia, eikä näitä ole mielekästä esitellä tämän enempää, koska TVD-menetelmillä on omat heikkoutensa. Sweby (1984, s. 1003) nostaa nimittäin

esille, että TVD-menetelmiä käytettäessä tulee täyttää Courant–Friedrichs–Lewy-ehto. Jasakin (1996, s. 100) mukaan tämä kuitenkin käytännössä tarkoittaa sitä, että käytetyn aika-askeleen suuruus voi vaikuttaa esimerkiksi stationääritilanjatkaisuun, joten parempana ratkaisuna on siirtyä käyttämään normalisoidun muuttujan menetelmää (engl. Normalised Variable Approach), eli lyhennettynä NVA.

Säännöllisen 1D-laskentaverkon tapauksessa mikä tahansa muuttuja ϕ voidaan normalisoida käyttäen ylävirran ja alavirran arvoja ϕ_U ja ϕ_D siten, että

$$\tilde{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U}. \quad (2.97)$$

Hyödyntämällä yleistä tapausta (2.97) voidaan kontrollitilavuuksien rajapinnan ϕ_f -arvo lausua normalisoidussa muodossa

$$\tilde{\phi}_f = \frac{\phi_f - \phi_U}{\phi_D - \phi_U}, \quad (2.98)$$

ja vastaavalla menettelyllä voidaan myös normalisoida kontrollitilavuuden keskipisteen arvo ϕ_C , jonka normalisoiduksi muodoksi saadaan

$$\tilde{\phi}_C = \frac{\phi_C - \phi_U}{\phi_D - \phi_U}. \quad (2.99)$$

NVA-menetelmän ideana on nyt rajoittaa koko laskentaverkossa ehto

$$\phi_U \leq \phi_C \leq \phi_D, \quad (2.100)$$

tai

$$\phi_U \geq \phi_C \geq \phi_D, \quad (2.101)$$

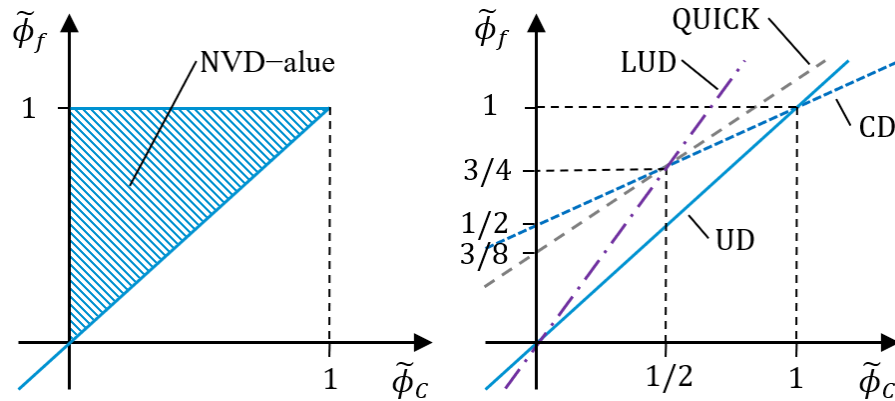
mikä varmistaa sen, ettei yksittäisen kontrollitilavuuden arvo ϕ_C voi kasvaa suuremmaksi tai pienemmäksi kuin sen ylä- ja alavirran naapureiden arvot ϕ_U ja ϕ_D . Hyödyntämällä ehtoa (2.100) yhtälöön (2.99) saadaan normalisoidulle muuttujalle $\tilde{\phi}_C$ ehdoksi

$$0 \leq \tilde{\phi}_C \leq 1, \quad (2.102)$$

jota yleensä kutsutaan konvektion rajoitusehdoksi (engl. Convection Boundedness Criterion), eli CBC-ehdoksi. Kyseinen ehto takaa, että konvektiomenetelmän tulokset pysyvät fysikaalisina, kun rajapinnan arvo $\tilde{\phi}_f$ muodostetaan käyttäen muuttujaa $\tilde{\phi}_C$. (Jasak 1996, s. 100–101)

CBC-ehto voidaan esittää graafisesti kuvan 2.15 tavalla NVD-kaavion (engl. Normalised Variable Diagram) muodossa. Mikäli perinteiset konvektiomenetelmät muutetaan NVD-

muotoon, huomataan, että vain ylävirran menetelmä toteuttaa CBC-ehdon, kuten kuvasta 2.15 voidaan päätellä. Muut menetelmät eivät kulje pisteiden (0,0) sekä (1,1) kautta, joten näitä ei voida pitää fysikaalisina. (Jasak 1996, s. 101–102) Oikealla kuvassa 2.15 aiemmista nimityksistä hieman poiketen UD vastaa ylävirran menetelmää, CD keskeisdifferenssimenetelmää, LUD lineaarista ylävirran menetelmää, ja QUICK aiemmin esillä ollut kvadraattista mallia normaalisti. Paneudutaan tarkemmin vielä siihen, miten NVA-viitekehystä voidaan hyödyntää mielivaltaisen muotoisille kontrollitulavuuksille.



Kuva 2.15 Vasemmalla CBC-rajoitusehto NVD-kuvaajassa ja oikealla yleisimmät konvektiomenetelmät NVD-viitekehyksessä. Mukailtu lähteestä (Jasak 1996, s. 102).

Ongelmaksi NVA-menetelmässä muodostuu yhtälöissä (2.98–2.99) esiintyvä ylävirran ϕ_U -arvo, jota usein ei epäsäännöllisen laskentaverkon tapauksessa tunneta. Koska laskentapiste C on kontrollitulavuuden keskellä, tehdään olettaus, että $\tilde{\phi}_C$ -arvo on välillä

$$\phi_f^- \leq \tilde{\phi}_C \leq \phi_f^+, \quad (2.103)$$

jossa ϕ_f^+ kuvaa rajapintojen arvoista suurempaa ja vastaavasti ϕ_f^- pienempää. Oletetaan ylävirran arvon olevan pienempi kuin alavirran arvo, jotta edellä mainitun avulla voidaan määrittää $\tilde{\phi}_C$ uudelleen. Kun korvataan $\phi_U = \phi_f^-$ ja $\phi_D = \phi_f^+$, normalisoidulle muuttujalle saadaan lauseke

$$\tilde{\phi}_C = \frac{\phi_C - \phi_f^-}{\phi_f^+ - \phi_f^-} = 1 - \frac{\phi_f^+ - \phi_C}{\phi_f^+ - \phi_f^-}, \quad (2.104)$$

joka ei sisällä enää lainkaan ylävirran tai alavirran arvoja. (Jasak 1996, s. 104–105)

Yhtälössä (2.104) esiintyviä rajapintojen arvoja ei kuitenkaan tunneta. Ratkaisuna ongelmaan on se, että arvojen $\phi_f^+ - \phi_C$ erotus lausutaan gradientin $\nabla\phi_f$ avulla muodossa

$$\phi_f^+ - \phi_C = g_C(\phi_D - \phi_C) = g_C \frac{(\phi_D - \phi_C)}{d_{DC}} d_{DC} = g_C d_{DC} (\nabla\phi)_f \cdot \mathbf{e}, \quad (2.105)$$

joka on saatu aikaan yksinkertaisella manipuloinnilla. Sijoittamalla saatu yhteys (2.105) yhtälöön (2.104) saadaan

$$\widetilde{\phi}_C = 1 - \frac{\phi_f^+ - \phi_C}{\phi_f^+ - \phi_f^-} = 1 - \frac{g_C d_{DC} (\nabla\phi)_f \cdot \mathbf{e}}{\phi_f^+ - \phi_f^-}. \quad (2.106)$$

Suorittamalla vastaavanlainen manipulaatio saadaan myös yhtälössä (2.106) tarvittava rajapintojen arvojen erotus $\phi_f^+ - \phi_f^-$ samanlaiseen gradienttimuotoon

$$\phi_f^+ - \phi_f^- = \frac{\phi_f^+ - \phi_f^-}{d_{f^+f^-}} d_{f^+f^-} = d_{f^+f^-} (\nabla\phi)_C \cdot \mathbf{e}. \quad (2.107)$$

Yhtälö (2.106) saadaan nyt yhteyden (2.107) avulla muotoon

$$\widetilde{\phi}_C = 1 - \frac{g_C d_{DC} (\nabla\phi)_f \cdot \mathbf{e}}{d_{f^+f^-} (\nabla\phi)_C \cdot \mathbf{e}}. \quad (2.108)$$

Oletetaan, että piste C on kontrollitilavuuden keskellä, minkä seurauksena interpolointikertoimelle pätee yhteys

$$g_C = \frac{\|\mathbf{x}_f^+ - \mathbf{x}_C\|}{\|\mathbf{x}_D - \mathbf{x}_C\|} = \frac{1}{2} \frac{\|\mathbf{x}_f^+ - \mathbf{x}_f^-\|}{\|\mathbf{x}_D - \mathbf{x}_C\|} = \frac{1}{2} \frac{d_{f^+f^-}}{d_{DC}}. \quad (2.109)$$

Sijoittamalla interpolaatiokertoimen tulos (2.109) yhtälöön (2.108) saadaan kontrollitilavuuden keskipisteessä tunnetulle normalisoidulle muuttujalle $\widetilde{\phi}_C$ lopullinen muoto

$$\widetilde{\phi}_C = 1 - \frac{(\nabla\phi)_f \cdot \mathbf{e}}{2(\nabla\phi)_C \cdot \mathbf{e}}, \quad (2.110)$$

joka ei riipu enää lainkaan ylävirran arvosta ϕ_U . Yhtälössä (2.110) esiintyvä $(\nabla\phi)_f \cdot \mathbf{e}$ tunnetaan, koska

$$(\nabla\phi)_f \cdot \mathbf{e} = \phi_D - \phi_C \quad (2.111)$$

Tulos (2.110) on voimassa kaikille laskentaverkkotyypeille, myös täysin epäsäännöllisille kontrollitilavuuksille, koska tämä ei enää riipu arvosta ϕ_U . (Jasak 1996, s. 104–107)

NVD-menetelmä osaa normalisoituun arvoon $\widetilde{\phi}_C$ perustuen valita, mitä konvektiomenetelmää se käyttää, mutta keskeisdifferenssimenetelmää kannattaa kuitenkin tarkkuutensa vuoksi hyödyntää mahdollisimman laaja-alaisesti. Sen sijaan tilanteissa, joissa $\widetilde{\phi}_C$ ei rajoitu CBC-ehdon (2.102) vaatimalle välille voidaan turvautua käyttämään ylävirran menetelmää, joka varmistaa numeerisen ratkaisun olevan edelleen fysikaalinen. Keskeisdifferenssimenetelmän ja ylävirran menetelmän välille tarvitaan jokin keino estää NVD-me-

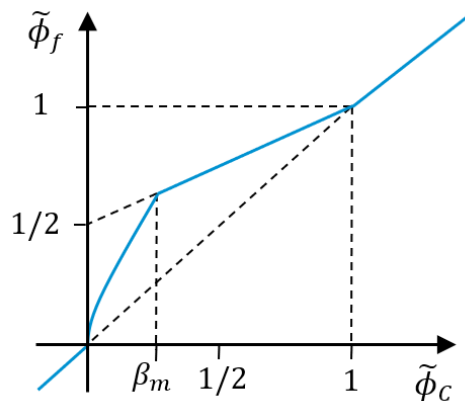
netelmän värähtelyt edellä mainittujen konvektiomenetelmien välillä. Tästä syystä on järkevää käyttää keskeisdifferenssimenetelmää vain johonkin alarajaan β_m asti. Transitioalueella $0 < \tilde{\phi}_c < \beta_m$ voidaan sekoittaa keskeisdifferenssimenetelmää ja lineaarista ylävirran menetelmää sekoituskertoimen $\gamma_m = \tilde{\phi}_c/\beta_m$ mukaisesti. Lopputuloksena saadaan NVD-menetelmä, joka osaa laskettujen gradienttien perusteella arvioida, mitä konvektiomenetelmää tulee soveltaa ja sen jälkeen määrittää rajapintojen arvot yhteyksistä (2.112). Kyseistä menetelmää voidaan kutsua Gamma-menetelmäksi. (Jasak 1996, s. 107–109) Työssä sovellettavaa Gamma-menetelmää on esitetty kuvassa 2.16.

$$\left\{ \begin{array}{ll} \tilde{\phi}_f = \tilde{\phi}_c, & 1 \leq \tilde{\phi}_c \\ \tilde{\phi}_f = \frac{1}{2} + \frac{1}{2}\tilde{\phi}_c, & \beta_m \leq \tilde{\phi}_c < 1 \\ \tilde{\phi}_f = -\frac{\tilde{\phi}_c^2}{2\beta_m} + \left(1 + \frac{1}{2\beta_m}\right)\tilde{\phi}_c, & 0 < \tilde{\phi}_c < \beta_m \\ \tilde{\phi}_f = \tilde{\phi}_c, & \tilde{\phi}_c \leq 0 \end{array} \right. \quad (2.112)$$

Rajoitusparametrin β_m valinnalla voidaan vaikuttaa Gamma-menetelmän toimivuuteen. Mikäli painotetaan menetelmän suppenemista stationääritilanteeseen, voidaan suosia suurempia β_m -arvoja. Parametrin lisääminen kuitenkin lisää numeerista diffuusiota, joten tälle kannattaa käyttää korkeintaan arvoa 0,5. Mikäli menetelmän halutaan toimivan epäjatkuvuuksien yhteydessä, parametrin arvo valitaan mahdollisimman pieneksi, arvon 0,1 tuntumaan. Edellä mainituista syistä rajoitusparametri β_m valitaan yleensä väliltä

$$\frac{1}{10} \leq \beta_m \leq \frac{1}{2}, \quad (2.113)$$

koska liian pienten rajoituskertoimien valitseminen saattaa johtaa tilanteisiin, joissa menetelmä vaihtelee konvektiomenetelmästä toiseen iteraatioiden välillä. Merkittävänä etuna Gamma-menetelmässä on se, että sen lopputulos ei riipu aika-askeleesta, eikä menetelmä siten kärsi aikariippuvuudesta. (Jasak 1996, s. 110–111)



Kuva 2.16 Gamma-menetelmä NVD-viitekehyksessä. Mukailtu lähteestä (Jasak 1996, s. 110).

2.6 Aikatermi

Tutustutaan vielä lyhyesti skalaarimuuttujan differentiaaliyhtälössä (2.1) esiintyvän aikatermin diskretisointiin. Samalla tavalla kuin muiden termien tapauksessa, myös aikatermille tarvitaan toisen kertaluvun diskretisointimenetelmä.

Kontrollitilavuusmenetelmällä ratkaistavien differentiaaliyhtälöiden aikatermin diskretisointia voidaan tutkia yksinkertaisen ensimmäisen kertaluvun alkuarvo-ongelman

$$\frac{d\phi(t)}{dt} = \hat{f}(t, \phi(t)) \quad (2.114)$$

avulla, minkä alkuehdoksi asetetaan $\phi(t_0) = \phi^0$. Integroimalla molemmat puolet yhtälöstä (2.114) ajan suhteen välillä $[t_n, t_{n+1}]$ saadaan ratkaisuksi

$$\phi^{n+1} - \phi^n = \int_{t_n}^{t_{n+1}} \hat{f}(t, \phi(t)) dt, \quad (2.115)$$

mutta ongelmaksi nousee, että yhtälön (2.115) oikean puolen integraalia ei voida laskea suoraan numeerisesti, ellei integraalia yksinkertaisteta jollakin menetelmällä. Yksinkertaisin tapa on olettaa $\hat{f}(t, \phi(t))$ vakioksi välillä $[t_n, t_{n+1}]$ ja määrätä sen suuruus ajanhetkellä t_n tunnetulla arvolla ϕ^n . Tämän seurauksena ajanhetkellä $\phi^{n+1} = \phi(t_{n+1})$ oleva uusi arvo saadaan yhteydestä

$$\phi^{n+1} = \phi^n + \hat{f}(t_n, \phi^n) \Delta t, \quad (2.116)$$

jota yleisesti kutsutaan eksplisiittiseksi Eulerin menetelmäksi. Vaihtoehtoisesti voidaan arvioida funktion $\hat{f}(t, \phi(t))$ suuruus ajanhetkellä t_{n+1} olevalla arvolla ϕ^{n+1} , jolloin muuttujan ϕ^{n+1} arvo määräytyy yhteydestä

$$\phi^{n+1} = \phi^n + \hat{f}(t_{n+1}, \phi^{n+1}) \Delta t. \quad (2.117)$$

Yhtälössä (2.117) tuntemattomana on molemmilla puolilla muuttuja ϕ^{n+1} , eli yhtälö täytyy vielä erikseen ratkaista, jos muuttujalle ϕ^{n+1} halutaan arvo ajanhetkelle t_{n+1} . Tästä syystä menetelmää kutsutaan implisiittiseksi Eulerin menetelmäksi. Kolmantena vaihtoehtona on hyödyntää funktion arvoa tunnetulla ajanhetkellä t_n ja tuntemattomalla ajanhetkellä t_{n+1} ja ottaa näistä aritmeettinen keskiarvo. Näin ollen saadaan Crankin–Nicolsonin menetelmä, eli

$$\phi^{n+1} = \phi^n + \frac{1}{2} [\hat{f}(t_n, \phi^n) + \hat{f}(t_{n+1}, \phi^{n+1})] \Delta t, \quad (2.118)$$

joka on myös implisiittinen. Yhtälö täytyy siis ratkaista, jotta saadaan ϕ^{n+1} -arvo tunnetuksi. Kyseisessä menetelmässä hyödynnettyä keskiarvoa kutsutaan myös puolisuunnikassäännöksi. (Ferziger & Perić 2002, s. 135–137)

Molempien Eulerin menetelmien voidaan osoittaa olevan vain ensimmäisen kertaluvun aikaintegrointimenetelmiä, eli niiden virhe riippuu lineaarisesti käytetyn aika-askeleen $\Delta t = t_{n+1} - t_n$ suuruudesta. Sen sijaan Crankin–Nicolsonin menetelmän voidaan osoittaa olevan toisen kertaluvun menetelmä, jolloin virhe pienenee $O(\Delta t^2)$ mukaisesti. Myös korkeamman kertaluvun menetelmiä voidaan johtaa, joista tyypillisimpiä ovat erilaiset Adamsin–Moultonin ja Rungen–Kuttan menetelmät. Toista kertalukua tarkemmat menetelmät vaativat poikkeuksetta yli kahden aika-askeleen arvot, joten näistä käytetään myös nimitystä monipistemenetelmät. (Ferziger & Perić 2002, s. 138–142)

Tässä työssä ei käydä läpi syvällisemmin aikaintegrointimenetelmiä tai pohdita niiden vaikutusta numeeriseen laskentaan, koska työn mielenkiinto on stationääritilan tapauksissa ja niiden numeerisessa tarkkuudessa. Myöhemmin esitettävä ratkaisualgoritmi voi käsitellä myös ajasta riippuvia virtaustilanteita, joten jonkinlainen käsitys aikatermin diskretisoinnista ei kuitenkaan ole pahitteeksi. Tarkempia tietoja aikaintegrointimenetelmistä löytyy lähteistä (Ferziger & Perić 2002) ja (Moukalled et al. 2016).

2.7 Lähde-termi

Kontrollitilavuusmenetelmässä lähde-termin diskretisointi on suoraviivaista. Oletetaan lähde-termin Q_C^ϕ riippuvan vain sen suuruudesta kontrollitilavuuden keskellä, eli

$$Q_C^\phi = Q(\phi_C), \quad (2.119)$$

josta voidaan lähde-termi arvioida sellaisenaan. Näin kannattaa tehdä vain hyvin yksinkertaisissa tapauksissa, jos lähde-termi on positiivinen. Jos lähde-termin merkki tunnetaan ja se on negatiivinen, eli lähde-termi on nielu, kannattaa se käsitellä implisiittisesti. Näin saadaan aikaiseksi laskentaa stabiloiva vaikutus. Lähde-termille $Q(\phi_C)$ on tapana tehdä Taylorin sarjakehitelmän tapainen linearisointi, jonka avulla päädytään yhtälöön

$$Q_C^{\phi, n+1} = Q(\phi_C^n) + \left(\frac{\partial Q}{\partial \phi_C} \right)^n (\phi_C^{n+1} - \phi_C^n) = \left(\frac{\partial Q}{\partial \phi_C} \right)^n \phi_C^{n+1} + Q(\phi_C^n) - \left(\frac{\partial Q}{\partial \phi_C} \right)^n \phi_C^n. \quad (2.120)$$

Yhtälössä (2.120) ensimmäinen termi muodostaa implisiittisen osuuden ja vastaavasti kaksi viimeistä termiä muodostavat eksplisiittisen osuuden, jotka arvioidaan edellisen iteraation n perusteella. Lähde-termin kontribuutio jokaisessa kontrollitilavuudessa on täten

$$\begin{aligned} Q_C^{\phi, n+1} V_C &= \iint_{V_C} Q^{\phi, n+1} dV = \iint_{V_C} \left(\frac{\partial Q_C^n}{\partial \phi_C} \phi_C^{n+1} \right) dV + \iint_{V_C} \left(Q_C^n - \frac{\partial Q_C^n}{\partial \phi_C} \phi_C^n \right) dV \\ &= \left(\frac{\partial Q_C^n}{\partial \phi_C} V_C \right) \phi_C^{n+1} + \left(Q_C^n - \frac{\partial Q_C^n}{\partial \phi_C} \phi_C^n \right) V_C, \end{aligned} \quad (2.121)$$

jonka implisiittinen osuus vähennetään kontrollitulavuuskohtaisen yhtälön a_C -osuudesta ja jäljelle jäävä eksplisiittinen osa kannattaa sisällyttää yhtälön oikealle puolelle b_C -kertoimeen. Tavoitteena on lisätä ratkaistavien yhtälöiden a_C -kertoimen suuruutta ja pyrkiä varmistamaan, etteivät nämä kertoimet mene negatiiviselle puolelle ja siten aiheuta divergenssiongelmaa. (Moukalled et al. 2016. s. 535–537)

2.8 SIMPLE- ja PISO-algoritmit sekä muut ratkaisumenetelmät

Soveltamalla edellä esitettyjä diskretisointimenetelmiä keskiarvoistettuihin Navierin–Stokesin yhtälöihin (2.21) ja turbulenssiyhtälöihin (2.24) ja (2.25) saadaan jokaiselle kontrollitulavuudelle yhteensä viisi ratkaistavaa yhtälöä. Tässä vaiheessa kannattaa huomata, että jatkuvuusehto (2.18) ei ole hyödynnetty. Tämän lisäksi liikemäärän säilymistä kuvaavat yhtälöt (2.21) riippuvat painegradientin suuruudesta. Edellä mainitusta syystä kontrollitulavuusmenetelmän yhteyteen on kehitetty useita menetelmiä, joilla tuntematon painegradientti saadaan selville ja jatkuvuusehto (2.18) saadaan täytettyä. Aiemmin esitettiin, kuinka differentiaaliyhtälöiden termien diskretisoinnin avulla saadaan selville kertoimet a_C , a_F ja b_C . Nämä voidaan nyt summata eri termien osalta ja muodostaa näiden avulla saatavista yhtälöistä yksi globaali yhtälöryhmä.

Likeyhtälöiden (2.21) tapauksessa jokaiselle komponentille saadaan lopulta muoto

$$a_C^{u_i} u_{i,C}^{n+1} + \sum_F^{n(F)} a_F^{u_i} u_{i,F}^{n+1} = b_{u_i}^{n+1} - \left(\frac{\partial p^{n+1}}{\partial x_i} \right)_C, \quad (2.122)$$

jossa tuntematon painegradientti on jätetty omaksi termikseen yhtälön oikealle puolelle. Yhtälöä (2.122) iteroitaessa vasemmalla puolella esiintyvät kertoimet $a_C^{u_i}$ ja $a_F^{u_i}$ sekä oikealla oleva $b_{u_i}^{n+1}$ lähdetermi voivat muuttua jokaisen iteraatiokierroksen välillä. Koska lähdetermi tai painegradientti eivät välttämättä ole tunnettuja, kannattaa yhtälö ratkaista edellisen iteraation tunnetuilla arvoilla. Tällöin ratkaistavaksi yhtälöksi muodostuu

$$a_C^{u_i} u_{i,C}^{m*} + \sum_F^{n(F)} a_F^{u_i} u_{i,F}^{m*} = b_{u_i}^{m-1} - \left(\frac{\partial p^{m-1}}{\partial x_i} \right)_C, \quad (2.123)$$

jossa merkittiin $u_{i,C}^{n+1} = u_{i,C}^{m*}$, ja tämä esittää yhtälön (2.123) ratkaisua iteraatiokierroksella m , eikä täten välttämättä kyseisen yhtälön lopullista ratkaisua. Yläindeksi vaihtui myös hieman, nyt $m = n + 1$. Yhtälö (2.123) ei täysin vastaa yhtälöä (2.122), koska oikealla puolella olevat termit on määritetty edellisen iteraation perusteella. Ratkaisemalla yhtälöstä (2.123) ainoa tuntematon $u_{i,C}^{m*}$ -termi, saadaan ratkaisu

$$u_{i,C}^{m*} = \frac{b_{u_i}^{m-1} - \sum_F^{n(F)} a_F^{u_i} u_{i,F}^{m*}}{a_C^{u_i}} - \frac{1}{a_C^{u_i}} \left(\frac{\partial p^{m-1}}{\partial x_i} \right)_C, \quad (2.124)$$

ja jatkoa ajatellen yhtälön (2.124) ensimmäiselle termille kannattaa käyttää lyhennettyä merkintää $\bar{u}_{i,C}^{m*}$. Täten yhtälölle (2.124) voidaan muodostaa kompaktimpi muoto

$$u_{i,C}^{m*} = \bar{u}_{i,C}^{m*} - \frac{1}{a_C^{u_i}} \left(\frac{\partial p^{m-1}}{\partial x_i} \right)_C, \quad (2.125)$$

mutta ongelmaksi nousee, että kyseinen ratkaisu ei välttämättä toteuta jatkuvuusyhtälöä

$$\frac{\partial(\rho u_i^m)}{\partial x_i} = 0, \quad (2.126)$$

mikä aiheuttaa sen, että yhtälön (2.125) ratkaiseminen monimutkaistuu. (Ferziger & Perić 2002, s. 172–174) Ongelma lisää ratkaistavien yhtälöiden määrää ja hankaloittaa laskentaa merkittävästi, kuten seuraavaksi tuodaan esille.

Ratkaisuna ongelmaan on se, että suoritetaan painegradientin avulla erillinen lisäkorjaus yhtälöstä (2.125) saatuun tulokseen, eli korjaus

$$u_{i,C}^m = \bar{u}_{i,C}^{m*} - \frac{1}{a_C^{u_i}} \left(\frac{\partial p^m}{\partial x_i} \right)_C, \quad (2.127)$$

joka varmistaa ratkaisun toteuttavan jatkuvuusyhtälön. Painegradientti on kuitenkin edelleen tuntematon. Sijoittamalla korjaus (2.127) jatkuvuusyhtälöön (2.126) saadaan muodostettua Poissonin yhtälö painegradientin laskentaa varten, eli yhtälö

$$\frac{\partial}{\partial x_i} \left[\frac{\rho}{a_C^{u_i}} \left(\frac{\partial p^m}{\partial x_i} \right) \right]_C = \left[\frac{\partial(\rho \bar{u}_i^{m*})}{\partial x_i} \right]_C, \quad (2.128)$$

ja ratkaisuna saadaan aiemmin tuntemattomana ollut paine. Sijoittamalla tästä ratkaistu painegradientti takaisin yhtälöön (2.127) saadaan lopullinen arvo nopeuksille $u_{i,C}^m$, mutta ratkaisu ei valitettavasti aina toteuta yhtälöä (2.123). Yhtälöitä (2.125), (2.127) ja (2.128) iteroidaan niin pitkään, kunnes alkuperäinen yhtälö (2.123) toteutuu. Samalla painegradientin korjaus varmistaa, että myös jatkuvuusehto (2.126) täytetään. Näitä iteraatioita yleisesti kutsutaan ulkoiteraatioksi. Tyypillisesti liikeyhtälöt voidaan ratkaista peräkkäin, koska jokaisessa yhtälössä on painegradientin lisäksi vain yksi tuntematon nopeuskomponentti. (Ferziger & Perić 2002, s. 174–175)

Yleisemmässä SIMPLE-algoritmissa (engl. Semi-Implicit Method for Pressure-Linked Equations) on tapana jakaa nopeusermi $u_{i,C}^m$ väliaikaiseen arvoon u_i^{m*} ja korjaukseen u' yhteyden

$$u_i^m = u_i^{m*} + u', \quad (2.129)$$

mukaisesti. Vastaavalla tavalla voidaan tehdä myös painetermille, eli muodostaa yhteys

$$p^m = p^{m-1} + p', \quad (2.130)$$

jossa p^{m-1} kuvaa paineen väliaikaista arvoa ja p' paineenkorjausta. Sijoittamalla edellä mainitut yhteydet (2.129) ja (2.130) yhtälöön (2.123) saadaan nopeuskorjaus lausuttua painekorjauksen avulla muodossa

$$u'_{i,c} = \bar{u}'_{i,c} - \frac{1}{a_c^{u_i}} \left(\frac{\partial p'}{\partial x_i} \right)_c, \quad (2.131)$$

jossa on merkitty

$$\bar{u}'_{i,c} = - \frac{\sum_F^{n(F)} a_F^{u_i} u'_{i,F}}{a_c^{u_i}}. \quad (2.132)$$

Samalla tavalla kuin aiemmin, painekorjatun nopeuskentän tulee toteuttaa jatkuvuusyhtälö. Jatkuvuusyhtälön toteutumisen takaa Poissonin yhtälön tapainen ehto

$$\frac{\partial}{\partial x_i} \left[\frac{\rho}{a_p^{u_i}} \left(\frac{\partial p'}{\partial x_i} \right) \right]_c = \left[\frac{\partial(\rho u_i^{m*})}{\partial x_i} \right]_c + \left[\frac{\partial(\rho u'_i)}{\partial x_i} \right]_c, \quad (2.133)$$

jossa yhtälön oikean puolen viimeinen termi on yleisesti tapana jättää SIMPLE-algoritmissa huomioimatta kokonaan, koska $u'_{i,c}$ ei ole tunnettu. Ad hoc -tyyppinen yksinkertaistus ei ole erityisen hyvä, sillä tämä laskee menetelmän suppenemista huomattavasti. (Ferziger & Perić 2002, s. 175)

Parempana vaihtoehtona on approksimoida $u'_{i,c}$ arvoja kontrollitilavuuden naapureiden painotettuna keskiarvona, eli

$$u'_{i,c} \approx \frac{\sum_F^{n(F)} a_F^{u_i} u'_{i,F}}{\sum_F^{n(F)} a_F^{u_i}}, \quad (2.134)$$

ja kyseisen oletuksen avulla saadaan approksimaatio $\bar{u}'_{i,c}$ -termille yhtälöstä (2.132), eli

$$\bar{u}'_{i,c} \approx -u'_{i,c} \frac{\sum_F^{n(F)} a_F^{u_i}}{a_c^{u_i}}. \quad (2.135)$$

Sijoittamalla approksimaatio (2.135) aiempaan painekorjausyhtälöön (2.131) saadaan nopeuskorjaukselle ja painekertoimelle tarkempi riippuvuus

$$u'_{i,c} = - \frac{1}{a_c^{u_i} + \sum_F^{nb(C)} a_F^{u_i}} \left(\frac{\partial p'}{\partial x_i} \right)_c, \quad (2.136)$$

jonka avulla voidaan johtaa Poissonin yhtälö muodossa

$$\frac{\partial}{\partial x_i} \left[\frac{\rho}{a_c^{u_i} + \sum_F^{n(F)} a_F^{u_i}} \left(\frac{\partial p'}{\partial x_i} \right) \right]_c = \left[\frac{\partial(\rho u_i^{m*})}{\partial x_i} \right]_c. \quad (2.137)$$

Näin johdettu algoritmi tunnetaan nimellä SIMPLEC (engl. Semi-Implicit Method for Pressure-Linked Equations Consistent), sillä kyseinen menetelmä on konsistenssi versio perinteisestä SIMPLE-algoritmista. Formulaatiosta on se merkittävä etu, ettei SIMPLEC vaadi lainkaan painekorjauksen eksplisiittistä alirelaksointia

$$p^m = p^{m-1} + \alpha^p p', \quad (2.138)$$

mitä joudutaan käyttämään SIMPLE-menetelmässä. SIMPLE-algoritmi saadaan kuitenkin suppenemaan lähes SIMPLEC-variantin nopeudella, kun alirelaksointikerroin α^p määritetään yhteydestä

$$\alpha^p = 1 + \frac{\sum_F^{n(F)} a_F^{u_i}}{a_p^{u_i}}. \quad (2.139)$$

Nopeudelle puolestaan saadaan alirelaksointikerroin α^U yksinkertaisesta kokemukseen perustuvasta ehdosta

$$\alpha^U + \alpha^p = 1, \quad (2.140)$$

mikä on kuitenkin yleisesti havaittu varsin hyvin toimivaksi säännöksi. (Ferziger & Perić 2002, s. 175–178)

Kolmantena vaihtoehtona liikeyhtälöiden ratkaisemiseen on lisätä käytettyjen korjausiteraatioiden määrää. Suoritetaan ensimmäinen korjausiteraatio SIMPLE-menetelmän mukaisesti, ja korjataan tämän jälkeen nopeudet vielä uudelleen yhteydestä

$$u_{i,c}'' = \bar{u}_{i,c}' - \frac{1}{a_c^{u_i}} \left(\frac{\partial p''}{\partial x_i} \right)_c, \quad (2.141)$$

jossa esiintyvä uusi painegradientti saadaan ratkaisemalla taas uusi Poissonin yhtälö

$$\frac{\partial}{\partial x_i} \left[\frac{\rho}{a_c^{u_i}} \left(\frac{\partial p''}{\partial x_i} \right) \right]_c = \left[\frac{\partial(\rho \bar{u}_i')}{\partial x_i} \right]_c, \quad (2.142)$$

joka varmistaa myös jatkuvuusehdon toteutumisen. Tämän jälkeen voidaan suorittaa kolmas nopeuskorjaus ja vastaavasti ratkaista myös kolmas Poissonin yhtälö ja niin edelleen, kunnes liikeyhtälöt on ratkaistu. Menetelmä tunnetaan yleisesti PISO-algoritmia (engl. Pressure-Implicit with Splitting of Operators), ja useimmiten menetelmässä käytetään vain kahta korjausiteraatiota. (Ferziger & Perić 2002, s. 176)

SIMPLE-algoritmissa tarvittavan eksplisiittisen alirelaksoinnin lisäksi voidaan joskus tarvita ratkaisuprosessin aikana alirelaksointia myös muille yhtälöille. Yleisesti käytetään Patankarin alirelaksoinnin periaatteena tunnettua menetelmää, minkä perusteella alirelaksoitu arvo määritetään yhteydestä

$$\phi_C^{n+1} = \phi_C^n + \alpha^\phi [\bar{\phi}_C^{n+1} - \phi_C^n]. \quad (2.143)$$

Tämän hetkinen $\bar{\phi}_C^{n+1}$ -arvo tunnetaan, joten implisiittistä alirelaksointia voidaan käyttää muodossa

$$\phi_C^{n+1} = \phi_C^n + \alpha^\phi \left[\left(\frac{-\sum_F^{n(F)} a_F \phi_F + b_C}{a_c} \right)^{n+1} - \phi_C^n \right], \quad (2.144)$$

jossa α^ϕ on jokin alirelaksointikerroin, jolla mielivaltaisen ϕ_C -muuttujan liian suurta vaihtelua iteraatiokertojen välillä pyritään hallitsemaan. Purkamalla yhtälö (2.144) auki muotoon

$$\frac{a_c}{\alpha^\phi} \phi_C^{n+1} + \sum_F^{n(F)} a_F \phi_F^{n+1} = b_C^{n+1} + \frac{(1 - \alpha^\phi)}{\alpha^\phi} a_c \phi_C^n \quad (2.145)$$

voidaan nähdä alirelaksointikertoimen skaalaavan vain diagonaalikerrointa a_c . Tällä saavutetaan se, että ratkaisuprosessin aikana yhtälöistä muodostettujen yhtälöryhmien kerroinmatriisin diagonaali muuttuu hallitsemammaksi ja samalla myös numeerinen stabiilius kasvaa. Vastaavanlaista hyötyä ei saavuteta eksplisiittisellä alirelaksoinnilla. Alirelaksoinnissa käytetään tyypillisesti kertoimia väliltä $[0,1]$, mutta alirelaksointimenetelmät eivät periaatteessa estä tätä suurempien kertoimien käyttöä. Tällöin alirelaksointi ei kuitenkaan enää kykene stabiloimaan laskentaa, eikä ratkaistavat yhtälöt välttämättä supene. (Moukalled et al. 2016, s. 539–541)

Nykyisin on tapana yhdistää PISO-algoritmi ja SIMPLE-algoritmi uudeksi PIMPLE-algoritmiksi (engl. merged PISO-SIMPLE). Menetelmällä voidaan käyttää merkittävän suuria aika-askeleita ja silti pitää laskenta stabiilina. Ideana on marsittaa aikaa eteenpäin ja ratkaista jokainen aika-askel siten, että saavutetaan stationääritila. Algoritmi toimii myös SIMPLEC-variantilla. Menetelmä voi mahdollistaa lähes kertaluokkaa suuremman aika-askeleen käytön, mutta liian suurien aika-askeleiden käyttäminen voi olla epäedullista tapauksissa, joissa virtauksen aikahistorialla on merkitystä. Oikein käytettynä PIMPLE-algoritmi voi kuitenkin ratkaista monimutkaisia virtaustapauksia huomattavasti paljon tehokkaammin kuin esimerkiksi PISO-algoritmi, koska PISO-algoritmi voi vaatia näissä tilanteissa huomattavan pienen aika-askeleen ollakseen stabiili. (Holzmann 2018, s. 92–111) Menetelmään paneudutaan tarkemmin neljännessä luvussa, kun PIMPLE-algoritmia sovelletaan tähän työhön.

3. NAVIERIN–STOKESIN YHTÄLÖIDEN VIRHEARVIOMENETELMÄT

3.1 Virhearviomenetelmien perusta

Edellisessä luvussa käsiteltiin erilaisten konvektio-, gradientti- ja diffuusiotermien diskretisointimenetelmiä. Menetelmien kokonaisvirheestä saatiin vain hyvin karkeita arvioita ja saatiin johdettua erilaisia kontrollitulavuuksien geometriasta johtuvia virhelähteitä. Ongelmana yksittäisten menetelmien tarkastelussa on se, että menetelmillä diskretisoitujen Navierin–Stokesin yhtälöiden (2.21) numeerisen ratkaisun todellista virhettä ei voida johtaa, eikä kontrollitulavuuskohtaisia virheitä tunneta. Sen sijaan elementtimenetelmän puolella on johdettu useita päteviä menetelmiä todellisten virheiden arviointiin ja elementtimenetelmälle johdetut virhearviot ovat siellä jo yleisesti käytössä (Ainsworth & Oden 1997, s. 3). Tästä syystä tässä työssä esitetään erilaisia virhearviomenetelmiä yleisellä tasolla elementtimenetelmän puolelta ja myöhemmissä alaluvuissa sovelletaan näitä kontrollitulavuusmenetelmän viitekehukseen. Ennen virhearvioiden käsittelyä kannattaa kuitenkin tehdä selkeä ero virhearvioiden ja esimerkiksi virheindikaattoreihin välille.

Adaptiivisessa kontrollitulavuusmenetelmässä yleensä sovelletaan virheindikaattoreita, joiden muodostamisessa käytetään helposti virtauskentästä saatavia suureita, kuten esimerkiksi nopeusgradientteja. Virheindikaattorien käyttäminen on hyväksyttävää esimerkiksi niissä tapauksissa, joissa virtauskentässä on luonnostaan epäjatkuvuuksia, kuten shokkiaaltoja. Epäjatkuvuuksia sisältävissä virtaustapauksissa tulosten tarkkuutena voidaan nimittäin pitää niissä esiintyvien epäjatkuvuuksien paksuutta, koska todellisuudessa nämä ovat on äärettömän ohuita. Sen sijaan, jos virtauskenttä on sileä, eikä sisällä edellä mainittuja epäjatkuvuuksia, ei virheindikaattoreita voida käyttää hyvällä menestyksellä. Syynä tähän on se, että sileästä virtauskentästä ei voida etukäteen tehdä päätelmiä, että riippuko numeerisen ratkaisun tarkkuus suoraan esimerkiksi nopeusgradienttien suuruudesta, eivätkä virheindikaattorit täten voi kuvata virheen todellista suuruutta näissä tapauksissa. Virheindikaattoreita voidaan näin ollen käyttää vain hyvin rajallisesti, koska nämä vain enimmäkseen tuovat esille sellaisia numeerisen ratkaisun osa-alueita, joissa adaptiivisuutta kannattaisi hyödyntää. (Jasak 1996, s. 154–155)

Mekaniikassa diskretisointivirheen arvioiminen on yleisesti johtanut *a posteriori* -virhearvioiden eli jälkikäteisvirhearvioiden syntymiseen ja näiden kehittämiseen erilaisille osittaisdifferentiaaliyhtälöille. Jälkikäteisvirhearvioissa voidaan hyödyntää jo valmiiksi laskettua numeerista ratkaisua ja sen jälkeen muodostaa tälle ratkaisulle virhearvio, joka osoittaa jollakin tavalla tämän ratkaisun tarkkuutta. Muodostetun jälkikäteisvirhearvion avulla voidaan parantaa numeerisen ratkaisun tarkkuutta, koska virhearvio osoittaa selke-

ästi, miten virhettä kannattaa lähteä pienentämään. Numeerisille ratkaisuille on tyypillistä, että nämä sisältävät aina jonkin määrän diskretisointivirhettä, eivätkä kykene kuvaamaan ratkaistavia yhtälöitä täysin tarkasti. Perinteisesti numeerisessa laskennassa on tästä syystä käytetty *a priori* -virhearvioita näiden virheiden arviointiin, eli etukäteisvirhearvioita, joiden ideana on osoittaa, miten nopeasti numeerisen ratkaisun virheet pienevät. Toisaalta virhearvioina on historiallisesti käytetty esimerkiksi asiantuntijan kokemusta tai muita subjektiivisia arvioita ratkaisun tarkkuudesta. Valitettavasti näitä ei voida pitää hyvinä menetelminä, vaan virhettä on pyrittävä arvioimaan täsmällisesti. Näin ollen päästään adaptiivisuuden käsitteeseen, eli algoritmiseen numeerisen ratkaisun parantamiseen jälkikäteisvirhearvioiden avulla. (Ainsworth & Oden 1997, s. 1–2) Adaptiivisuuden käytännön toteuttamista käsitellään tarkemmin neljännessä luvussa, mutta toistaiseksi lukijalle riittää tieto, että jälkikäteisvirhearvion avulla parannetaan käytettyä laskentaverkkoa muuttamalla sen tiheyttä virhearvion osoittamissa paikoissa.

Seuraavaksi esitetään muutamia keskeisiä havaintoja elementtimenetelmään johdetuista jälkikäteisvirhearvioista. Tämä tehdään sen vuoksi, että esimerkiksi Carstensen et al. (2005, s. 2497) mukaan kontrollitulavuusmenetelmän virhearvioiden teoriapohjaa ei voida pitää vielä täysin valmiina, vaikka elementtimenetelmän virhearvioiden teoriapuolelta on saatu lainattua luotettavia menetelmiä myös virtausmekaniikan tarpeisiin. Sen sijaan (Ainsworth & Oden 1997, s. 3) toteaa elementtimenetelmän virhearvioiden teorian olevan jo niin pitkälle kehitetty, että painopiste on enemmän virhearviomenetelmien suorituskyvyn tutkimisessa kuin kokonaan uudenlaisten virhearvioiden kehittämisessä. Carstensen et al. (2005, s. 2498) myös huomauttaa, että kontrollitulavuusmenetelmän ja elementtimenetelmän ero on matemaattisesti loppujen lopuksi varsin pieni, joten elementtimenetelmän puolelta voidaan soveltaa virhearviomenetelmiä ilman sen kummempia ongelmia. Merkitään virhearvioissa Ainsworth & Oden (1997, s. 3–5) mukaisesti laskentaaluetta symbolilla Ω , sen reunaa $\partial\Omega$ ja lokaalia elementtikokoa h_C . Lisäksi Ω_C viittaa yksittäiseen elementtiin, $\partial\Omega_C$ elementin reunoihin ja γ elementtien välillä oleviin reunoihin.

Elementtimenetelmään johdetut jälkikäteisvirhearviot voidaan karkeasti jakaa kolmeen eri ryhmään. Yksinkertaisimpina näistä ovat niin sanotut gradienttimenetelmät, joissa verrataan lasketun gradientin arvoa tarkkana pidetyn gradientin suuruuteen. Toisena vaihtoehtona on määrittää jokaisen elementin sisältämät residuaalit, eli jäännökset, jotka osoittavat, että numeerinen ratkaisu ei täytä alkuperäistä ongelmaa täysin. Integroimalla nämä jäännökset jokaisen elementin yli saadaan muodostettua käyttökelpoinen elementtikohtainen virhe-estimaatti, joka kuvaa kyseisessä elementissä esiintyvää diskretisointivirhettä. Yleisesti kyseinen jäännös menetelmä luokitellaan kuuluvan eksplisiittisiin virhearviointeihin. Kolmantena vaihtoehtona on hyödyntää implisiittisiä jälkikäteisvirhearvioita, joissa ratkaistaan jokaista elementtiä kohden saatava jäännösongelma. Implisiitti-

sestä virhearviosta on se etu eksplisiittiseen virhearvioon nähden, että tällä saadaan johdettua luotettava yläraja diskretisointivirheelle. Yhteistä kaikille näille menetelmille on kuitenkin niistä saatava globaali virhe-estimaatti

$$\eta = \left(\sum_{C=1}^{n(C)} \eta_C^2 \right)^{1/2}, \quad (3.1)$$

jossa η_C on jokaisen elementin C lokaali virhe-estimaatti. Kaikille virhearvoille on vaatimuksena, että globaalille virhearviolle η voidaan johtaa yläraja ja alaraja

$$C_1 \|e\|_E \leq \eta \leq C_2 \|e\|_E, \quad (3.2)$$

jossa C_1 ja C_2 ovat positiivisia vakioita ja $\|e\|_E$ kuvastaa diskretisointivirheelle

$$e = u - u_h \quad (3.3)$$

globaalisti määritettyä energianormia. Toinen tärkeä ehto virhearvioille on se, että globaali tehokkuussuhde

$$\frac{\eta}{\|e\|_E} \quad (3.4)$$

on mahdollisimman lähellä arvoa 1, mikä kertoo virhe-estimaatin η suppenevan samalla nopeudella, kuin todellinen virhe. Vaatimus ei kuitenkaan ole ehdoton, sillä kyseinen suhde voi ylittyä ja estimaatti voi silti olla käyttökelpoinen. Tyypillisesti jopa yli 3,0 efektiivisyysuhde voidaan sallia käytännön insinöörisovelluksissa. (Ainsworth & Oden 1997, s. 7–9) Ennen kuin virhe-estimaatteja formuloidaan kontrollitulavuusmenetelmään, selvitetään, mitä erityyppiset virhe-estimaatit tarkalleen ottaen pitävät sisällään.

Gradientteja hyödyntävissä menetelmissä ajatellaan numeerisen ratkaisun gradienttien vaikuttavan eniten kokonaisvirheeseen. Tämän seurauksena vain gradienttien aiheuttama virhettä kannattaa lähteä arvioimaan. Tarkastellaan skalaariongelmaa

$$-\nabla \cdot (\nabla u) + cu = \hat{f}, \quad (3.5)$$

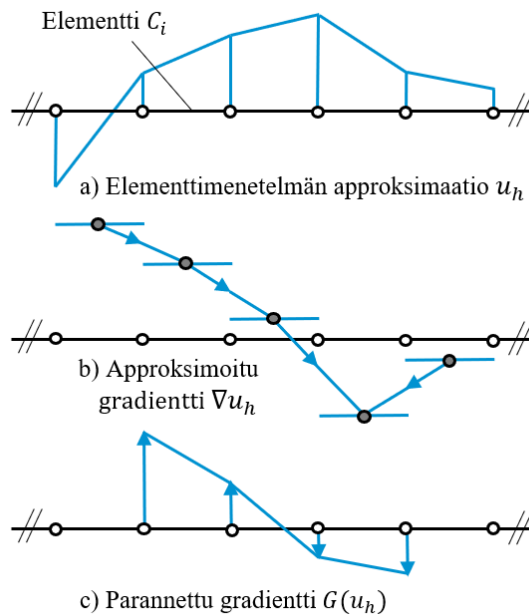
jonka tapauksessa todellinen diskretisointivirhe saadaan yhteydestä

$$\|e\|_E = \int_{\Omega} |\nabla u - \nabla u_h|^2 dV, \quad (3.6)$$

jossa ∇u on todellinen gradientti ja ∇u_h numeerisesta ratkaisusta saatu gradientti. Ongelmaksi nousee se, että todellinen gradientti on miltei aina tuntematon. Sovittamalla jatkuva gradienttijakauma laskenta-alueelle Ω saadaan approksimoitua todellista gradienttia esimerkiksi operaattorilla

$$G_h(\Pi_s u) = \nabla u, \quad (3.7)$$

jossa Π_s interpoloi laskenta-alueen sisällä kertaluvun s tarkkuudella. Tavoitteena on etsiä sellainen $G_h(\Pi_s u)$ -operaattori, joka muodostaa saatavilla olevasta datasta sileän gradienttijakauman, jonka voidaan olettaa olevan tarpeeksi lähellä todellista gradienttia. Yleisesti myös vaaditaan, että operaattorissa esiintyvä Π_s interpoloi kertaluvun s tarkkuudella, jos numeerinen ratkaisu on diskretisoitu käyttäen kertaluvun $s + 1$ menetelmiä. Toinen selkeä vaatimus on, että $G_h(\Pi_s u)$ on rajoitettu, eikä kyseisellä operaattorilla parannettu gradientti voi kasvaa ympäröiviä solmukohtien gradientteja liioin suuremmaksi. Gradienttien ratkaiseminen ei voi myöskään vaatia globaalien ongelmien ratkaisemista, vaan se tulee olla muodostettavissa lokaaleiden arvojen avulla. (Ainsworth & Oden 1997, s. 9–14)



Kuva 3.1 Parannetun gradientin menetelmä. Mukailtu lähteestä (Ainsworth & Oden 1997, s. 17).

Edellä mainitut ehdot toteutuvat esimerkiksi 1D-tapauksessa, kun jokaisen elementin keskimääräisestä gradientista muokataan operaattorilla $G_h(\Pi_s u)$ paloittain jatkuva lineaarinen gradienttijakauma. Tällöin elementtikohtainen virhearvio saadaan vertaamalla paloittain jatkuvaa gradienttijakaumaa alkuperäiseen gradienttijakaumaan, eli yhteydestä

$$\eta_{G,C}^2 = \|G_h(u_h) - \nabla u_h\|_{L_2(\Omega_C)}, \quad (3.8)$$

jossa $\|\cdot\|_{L_2(\cdot)}$ kuvaa neliöintegroituvan funktion L_2 -normia. Yleisesti tunnettuna heikkoutena gradienttimenetelmillä on kuitenkin se, että ne toteuttavat virhe-estimaateilta vaaditun asymptoottisuus ominaisuuden

$$\lim_{h \rightarrow 0} \frac{\eta}{\|e\|_E} = 1 \quad (3.9)$$

vain hyvin rajoitetuissa tapauksissa. Kyseisissä tapauksissa toteutuu niin sanottu superkonvergenssiehto. Tällöin parannetut gradientit on saatu määritettyä esimerkiksi toisen kertaluvun tarkkuudella, vaikka näiden tulisi luonnostaan olla vain ensimmäistä kertalukua. Kyseinen ilmiö esiintyy esimerkiksi 1D-tapauksessa, kun käytetään paloittain lineaarista interpolointia ja 2D-tapauksissa, jos käytetään bilineaarista interpolointia jatkuvan gradienttijakauman muodostamisessa. Superkonvergenssi on kuitenkin erittäin harvainen ominaisuus ja tämä täytyy varmistaa tapauskohtaisesti. (Ainsworth & Oden 1997, s. 15–19) Edellä mainittua paloittain lineaarista 1D-tapauksen interpolointimetodia on esitetty kuvassa 3.1, joka myös täyttää edellä mainitun superkonvergenssiehdon.

Eksplisiittisissä virhearviomenetelmissä haetaan diskretisointivirheen energianormille $\|e\|_E$ virhe-estimaattia prosessoimalla jo valmiiksi saatavilla olevasta numeerisesta ratkaisusta siinä esiintyvät diskretisointivirheet. Esimerkiksi malliongelman (3.5) tapauksessa voidaan numeerisesta ratkaisusta havaita elementtikohtaisesti jäännös

$$r = \hat{f} + \nabla \cdot (\nabla u_h) - cu_h, \quad \text{alueella} \quad \Omega_C, \quad (3.10)$$

ja reunamilla, joille on määrätty malliongelmassa luonnollinen reunaehto \hat{g} , voidaan todeta vuojaäännös

$$R = \hat{g} - \frac{\partial u_h}{\partial n}, \quad \text{reunalla} \quad \partial\Omega_C \cap \Gamma_N, \quad (3.11)$$

missä \mathbf{n} kuvaa elementin reunapinnan pinnan normaalin suuntaista yksikkövektoria. Integroimalla jäännökset (3.10) ja (3.11) jokaisen elementin yli, saadaan jokaiselle elementille lokaali eksplisiittinen virhe-estimaatti

$$\eta_{E,C}^2 = h_C^2 \|r\|_{L_2(\Omega_C)}^2 + \frac{1}{2} h_C \|R\|_{L_2(\partial\Omega_C)}^2, \quad (3.12)$$

joiden globaalina summana saadaan yläraja energianormille $\|e\|_E$ muodossa

$$\|e\|_E \leq C_0 \sum_{C=1}^{n(C)} \left(h_C^2 \|r\|_{L_2(\Omega_C)}^2 + \frac{1}{2} h_C \|R\|_{L_2(\partial\Omega_C)}^2 \right), \quad (3.13)$$

jossa esiintyvä C_0 kerroin on tuntematon. (Ainsworth & Oden 1997, s. 21–23) Jos verrataan eksplisiittistä menetelmää gradienttimenetelmään, voidaan huomata, että eksplisiittisellä virhearviolla ei välttämättä päästy yhtään sen lähemmäksi luotettavaa diskretisointivirheen ylärajaa. On siis varmistettava joko superkonvergenssiehto erilaisia jatkuvia gradienttijakaumia muodostettaessa tai määritettävä tuntematon C_0 -kerroin, mutta molemmat vaihtoehdot ovat työläitä.

Todellisuudessa jäännösten (3.10) ja (3.11) arvioinnissa tehdään virhettä, koska funktio \hat{f} joudutaan interpoloimaan elementtien sisällä ja funktio \hat{g} elementtien reunamilla, joille

on asetettu luonnolliset reunaehdot. Tästä syystä eksplisiittiselle virhe-estimaatille saadaan tarkempi raja

$$\frac{1}{C_0} \|e\|_E^2 \leq \sum_{C=1}^{n(C)} \eta_C^2 \leq C_0 \left(\|e\|_E^2 + \sum_{C=1}^{n(C)} h_C^2 \|\hat{f} - \Pi_S \hat{f}\|_{L_2(\Omega_C)}^2 + \sum_{\gamma=1}^{n(\gamma)} h_C \|\hat{g} - \Pi_S \hat{g}\|_{L_2(\gamma)}^2 \right), \quad (3.14)$$

mutta käytännössä interpoloinnissa tehty virhe on merkityksetön. Myös numeerisen integroinnin vaikutus voidaan ottaa huomioon, mutta olettamalla numeerisen datan olevan tarpeeksi sileä, voidaan yksinkertaistaa eksplisiittistä virhearviota (3.14) muotoon

$$\frac{1}{C_0} \|e\|_E^2 \leq \sum_{C=1}^{n(C)} \eta_C^2 \leq C_0 \|e\|_E^2, \quad (3.15)$$

jossa esiintyvä lokaali virhe-estimaatti η_C^2 mahdollistaa esimerkiksi adaptiivisten algoritmien käyttämisen, koska diskretisointivirhe tunnetaan nyt elementtikohtaisesti, eikä vain globaalisti. Menettelyllä saatua virhearviota kutsutaan yleisesti eksplisiittiseksi virhearvioksi, koska kaikki globaalia virhearviota (3.15) varten tarvittavat termit tunnetaan. Rajojen sisältämä kerroin C_0 on sen sijaan riippuvainen elementtien säännöllisyydestä, muttei suoranaisesti muuttujista u tai h . (Ainsworth & Oden 1997, s. 26–29) Jäännöspohjainen eksplisiittinen virhearvio kontrollitilavuusmenetelmän viitekehyksessä esitetään tarkemmin luvussa 3.4.

Gradienttimenetelmien ja eksplisiittisten virhearvioiden lisäksi saadaan mielenkiintoinen jälkikäteisarvio, jos diskretisointivirheelle muodostetaan alkuperäisen malliongelman (3.5) avulla yhtälö

$$\nabla \cdot (\nabla e) + ce = \hat{f} + \nabla \cdot (\nabla u_h) - cu_h, \quad (3.16)$$

jonka ratkaisuna saadaan selville jokaisen elementin diskretisointivirhe. Tähän vaaditaan kuitenkin tätä ennen asianmukaiset reunaehdot, jotta yhtälö (3.16) voidaan ylipäätään ratkaista elementtikohtaisesti. Jos elementille on alun perin annettu oleelliset reunaehdot, annetaan vastaavasti diskretisointivirheelle oleellinen homogeeninen reunaehto

$$e = 0, \quad \text{reunalla} \quad \partial\Omega_C \cap \Gamma_D, \quad (3.17)$$

eli tällöin oletetaan, että numeerinen ratkaisu täyttää aina sille asetetut oleelliset reunaehdot. Vastaavasti malliongelmassa luonnollisia reunaehtoja käyttäville elementeille annetaan reunaehto

$$\frac{\partial e}{\partial n} = \hat{g} - \frac{\partial u_h}{\partial n}, \quad \text{reunalla} \quad \partial\Omega_C \cap \Gamma_N, \quad (3.18)$$

mutta ongelmaksi nousee se, miten laskenta-alueen sisällä annetaan reunaehdot, jos yhtälö (3.16) halutaan ratkaista elementtikohtaisesti, koska kaikki elementit eivät välttämättä noudata reunaehtoja (3.17) tai (3.18). Asetetaan laskenta-alueen sisälle elementtien välisille rajapinnoille reunaehdot

$$\frac{\partial e}{\partial n} = \frac{\partial u}{\partial n} - \left(\frac{\partial u_h}{\partial n} \right)_c, \quad (3.19)$$

ja yhteydessä (3.19) esiintyvä tarkka vuo voidaan arvioida kahden elementin gradientin keskiarvon avulla, eli approksimatiivisesti

$$\frac{\partial u}{\partial n} \approx \frac{1}{2} [(\nabla u_h)_C + (\nabla u_h)_F] \cdot \mathbf{n}, \quad (3.20)$$

mutta formuloituun ongelmaan ei kuitenkaan aina löydetä yksikäsitteistä ratkaisua edes näin alkeellisen malliongelman tapauksessa, koska yhtälön (3.16) ratkaisemisessa tarvittava, numeerisesta ratkaisusta saatu data, ei aina täytä niin sanottuja yhteensopivuusehtoja, ellei ongelmaan tehdä muutoksia. Valitsemalla elementtien välillä olevat vuot (3.20) hieman tarkemmin, saadaan tehokas jälkikäteisvirhearviomenetelmä, jota yleensä kutsutaan tasapainotettujen jäännösten menetelmäksi. Menetelmässä nimensä mukaisesti lokaalit ongelmat on tasapainotettu siten, että näille saadaan aina ratkaisu. Näin esitetystä implisiittisestä menetelmästä on se etu eksplisiittiseen menetelmään verrattuna, että diskretisointivirheelle saadaan tarkka yläraja muodossa

$$\|e\|_E^2 \leq \sum_{C=1}^{n(C)} \|\varphi\|_{E,C}^2, \quad (3.21)$$

jossa φ_C on implisiittinen lokaali virhe-estimaattori, eikä diskretisointivirheen yläraja enää riipu tuntemattomasta vakiosta C_0 . (Ainsworth & Oden 1997, s. 32–69) Tasapainotettujen jäännösten menetelmällä saatavaan virhearvioon perehdytään tarkemmin luvussa 3.5, jossa tätä myös sovelletaan Navierin–Stokesin yhtälöihin.

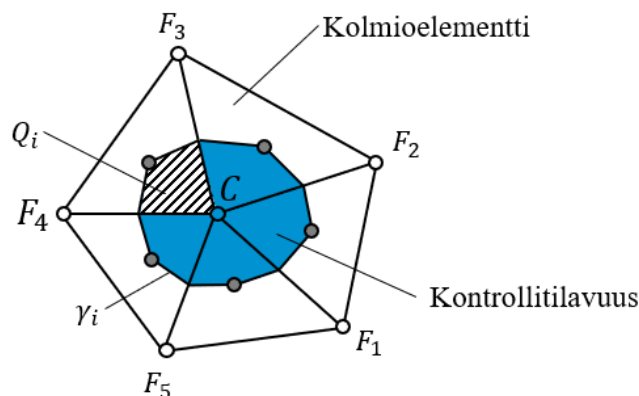
Edellä esitettyjen virhearviotyyppien yhteisenä piirteenä on se, että energianormia $\|e\|_E^2$ pyritään approksimoimaan siten, että itse energianormia ei tarvitse laskea. Todellinen diskretisointi virhe harvoin tunnetaan, eikä tarkkaa ratkaisua differentiaaliyhtälöihin ole useinkaan saatavilla. Edellä esitettyjen virhearvioiden ylärajojen ja alarajojen pätevyyttä ei sen kummemmin todistettu, mutta asiasta kiinnostuneet voivat tutustua tarkemmin lähteeseen (Ainsworth & Oden 1997), joka sisältää funktionaalianalyysiin pohjautuvaan todistukset näille rajoille.

Elementtimenetelmän puolella johdetut tulokset ovat siinä mielessä mielenkiintoisia, että duaalimenetelmällä on saatu johdettua vastaavanlaisia eksplisiittisiä virhe-estimaatteja. Esimerkiksi Afif et al. (2003, s. 130) on johtanut hyvin käyttökelpoisen eksplisiittisen virhearvion elliptiselle 2D-malliongelmalle muodossa

$$\eta_C^2 = h_C^2 \sum_{Q=1}^{n(Q)} \|r_C\|_{L_2(\Omega_C),Q}^2 + \frac{1}{2} \sum_{\gamma=1}^{n(\gamma)} h_\gamma \|\llbracket \nabla u_h \cdot n \rrbracket_\gamma\|_{L_2(\partial\Omega_C),\gamma}^2 \quad (3.22)$$

jossa jäännökset r_C summataan aina jokaisen monikulmion muotoisen kontrollitulavuuden Q yli ja vastaavasti vuo­jäännökset, eli gradientti­hyppäykset, monikulmioiden reunamien γ yli. Tulos (3.22) on siinä mielessä varsin rohkaiseva, että duaalimenetelmällä on saatu johdettua yksittäiselle kontrollitulavuudelle virhearvio, jos ajatellaan, että jokainen kontrollitulavuus koostuu useista pienemmistä kontrollitulavuuksia. Toinen hyvä ominaisuus virhearviossa (3.22) on se, että se vastaa elementtimenetelmän puolella johdettua tulosta (3.13). Tässä työssä etsitään kuitenkin perinteiselle kontrollitulavuusmenetelmälle hel­posti sovellettavia virhearvioita, joten duaalimenetelmälle johdettu virhe-estimaatti (3.22) ikävä kyllä joudutaan sivuuttamaan, koska tätä ei voida suoraan hyödyntää, sillä jäännökset r_C eivät ole suoraan saatavilla, kuten duaalimenetelmässä.

Duaalimenetelmä on Ferziger & Perić (2002, s. 245–246) mukaan yhdistelmä elementti­menetelmää ja kontrollitulavuusmenetelmää, missä esimerkiksi 2D-tapauksessa kolmioi­dusta laskenta-alueesta saadaan elementtien keskipisteitä ja reunojen keskipisteitä sopi­vasti keskenään yhdistelemällä monikulmion muotoisia kontrollitulavuuksia. Kuvassa 3.2 on esitetty tyypillinen duaalimenetelmässä käytetty 2D-laskentaverkko.



Kuva 3.2 Tyypillinen elementtimenetelmän ja kontrollitulavuusmenetelmän duaalimenetelmässä käytetty laskentaverkko. Mukailtu lähteestä (Ferziger & Perić 2002, s. 245).

Seuraavaksi luvussa 3.2 johdetaan helppokäyttöisiä virhe-estimaatteja ja luvuissa 3.3–3.4 keskitytään puhtaasti kontrollitulavuusmenetelmään johdettuihin eksplisiittisiin virhe-estimaatteihin. Näiden jälkeen siirrytään huomattavasti monimutkaisempaan implisiittiseen virhearviomenetelmään luvussa 3.5 ja samalla myös saadaan tarkempi yläraja diskreti­sointivirheelle kuin eksplisiittisillä menetelmillä.

3.2 Diskretisointivirheen arvioiminen Taylorin sarjakehitelmän avulla

Oletetaan kontrollitulavuuksilla diskretisoidun laskenta-alueen kuvaavan riittävän hyvin todellista virtaustilannetta, minkä seurauksena yksittäisten kontrollitulavuuksien diskretisointivirhe voidaan ilmoittaa approksimatiivisesti Taylorin sarjakehitelmänä, eli muodossa

$$e_h \approx Dh^s + H, \quad (3.23)$$

jossa H kuvaa korkeamman asteen termejä ja D on tuntematon kontrollitulavuuden koosta h riippumaton kerroin. Ajatellaan, että mielivaltaisen ϕ -muuttujan tarkka arvo Φ saadaan jokaisessa kontrollitulavuudessa siten, että korjataan approksimaatiosta (3.23) saatavalla diskretisointivirheellä numeerisen ratkaisun arvoa ϕ_h . Tarkaksi arvoksi täten saadaan

$$\Phi = \phi_h + e_h = \phi_h + Dh^s + H, \quad (3.24)$$

ja toinen vastaavanlainen yhtälö saadaan, jos lasketaan sama ongelma tuplasti harvemmallalla tai tiheämmällä laskentaverkolla. Esimerkiksi harvemman laskentaverkon tapauksessa pätee

$$\Phi = \phi_{2h} + e_{2h} = \phi_{2h} + D(2h)^s + H, \quad (3.25)$$

jossa Φ on edelleen sama arvo kuin yhtälössä (3.24). Jätetään korkeamman asteen termit H huomioimatta yhtälöistä (3.24) ja (3.25), minkä seurauksena voidaan ratkaista tuntematon D -kerroin näistä yhtälöistä ja sijoittaa tämä yhtälöön (3.23). Lopputuloksena on diskretisointivirheelle saatava arvio

$$e_h \approx \frac{\phi_h - \phi_{2h}}{2^s - 1}, \quad (3.26)$$

jossa s kuvaa laskennassa käytetyn diskretisointimenetelmän kertalukua. Jos tämä on tuntematon, se voidaan määrittää yhteydestä

$$s = \frac{\log\left(\frac{\phi_{2h} - \phi_{4h}}{\phi_h - \phi_{2h}}\right)}{\log 2}. \quad (3.27)$$

Menetelmää yleisesti kutsutaan Richardsonin ekstrapoloinniksi, koska diskretisointivirheen e_h avulla voidaan ikään kuin ekstrapoloida tarkempi numeerinen ratkaisu kahden tunnetun numeerisen ratkaisun ϕ_{2h} ja ϕ_h avulla. (Ferziger & Perić 2002, s. 58–60)

Todellisuudessa laskentaverkon koon tuplaaminen tai puolittaminen on liian rajoittava vaatimus, joten Richardsonin ekstrapoloinnilla saatava diskretisointivirhe kannattaa yleistää muotoon

$$e_h \approx \frac{\phi_h - \phi_{2h}}{\lambda^s - 1}, \quad (3.28)$$

jossa

$$\lambda = \frac{h_2}{h_1} \quad (3.29)$$

kuvaa laskentaverkkojen koon suhdetta. Tulos (3.28) on siinä mielessä käyttökelpoisempi, että λ -suhde voi olla mielivaltainen, mutta edelleen kuitenkin vaaditaan, että laskentaverkkoa tihennetään samassa suhteessa jokaisessa suunnassa. Diskretisointivirheelle johdettua yhteyttä (3.28) voidaan käyttää myös Richardsonin virhe-estimaattina η_R muodossa

$$\eta_R = \frac{\epsilon^\phi}{\lambda^s - 1}, \quad (3.30)$$

jossa muutosuhde ϵ^ϕ muodostetaan tiheämmän laskentaverkon tuloksen ϕ_{h_1} ja harvemman laskentaverkon tuloksen ϕ_{h_2} avulla yhteydestä

$$\epsilon^\phi = \frac{\phi_{h_2} - \phi_{h_1}}{\phi_{h_1}}. \quad (3.31)$$

Estimaattia (3.30) voidaan pitää hyvänä vain niissä tapauksissa, joissa $\eta_R \ll 1$. Yleisesti virtausmekaniikassa jälkikäteisvirhearviona on käytetty edellä mainittua ϵ^ϕ -suhdetta, joka ei suoranaisesti riipu diskretisointimenetelmien kertaluvusta s tai laskentaverkkojen koon suhteesta λ , eikä estimaattia (3.31) voida siten pitää riittävän hyvänä. (Roache 1994, s. 406–407)

Richardsonin virhe-estimaatti ei kuitenkaan anna luotettavaa ylärajaa diskretisointivirheelle, mutta sen sijaan Richardsonin virhe-estimaatin avulla voidaan muodostaa numeeristen laskentatulosten raportoinnin kannalta varsin hyödyllinen diskretisointivirheen suppenemista kuvaava GCI-indeksi (engl. Grid Convergence Index). Indeksia voidaan määrittää yhteydestä

$$\eta_{GCI} = \frac{3|\epsilon^\phi|}{\lambda^s - 1}. \quad (3.32)$$

Indeksin perimmäisenä tarkoituksena on normalisoida kaikki erilaisilla λ - ja s -parametreilla tehdyt virhe-estimaatit, mutta indeksissä esiintyvä kerroin on puhtaasti ad hoc -tyyppisesti määritetty, jolla pyritään saamaan indeksistä konservatiivinen. Richardsonin ekstrapolointi kuitenkin vaatii tulosten virheen pienenevän asymptoottisesti, minkä toteutamiseen tarvitaan usein vähintään kolme eri numeerista ratkaisua eri laskentaverkoilla. (Roache 1994, s. 407–411)

Richardsonin virhe-estimaatti on kuitenkin alkeellinen, jos verrataan tätä edellisessä luvussa esitettyihin eksplisiittisiin ja implisiittisiin virhearvioihin. Kolmen eri laskentaverkon käyttäminen on varsin työlästä ja muutenkin virhearvio jättää korkeamman asteen virhelähteet huomioimatta täysin. Tästä syystä Richardsonin virhe-estimaattia ei hyödynnetä tässä työssä, mutta menetelmällä saadaan jonkinlainen kontrollitulavuuskohtainen virhe-estimaatti, mikä on ehdottomasti jo iso askel oikeaan suuntaan. Tutustutaan seuraavaksi hieman monimutkaisempaan eksplisiittiseen virhearvioon.

3.3 Momenttimenetelmä

Kontrollitulavuusmenetelmässä numeerista ratkaisua voidaan pitää tarkkana, jos kontrollitulavuuksien keskipisteissä ja kontrollitulavuuksien rajapinnoille interpoloidut arvot vastaavat analyttistä ratkaisua. Mielivaltaisen muuttujan ϕ differentiaaliyhtälön

$$\nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\Gamma \phi \nabla \phi) + Q \phi \quad (3.33)$$

tapauksessa myös kaikki ratkaisun ϕ korkeammat momentit toteuttavat kyseisen yhtälön, mutta vain siinä tapauksessa, jos numeerinen ratkaisu on tarkka. Esimerkiksi muuttujan ϕ toisen momentin

$$m^\phi = \frac{1}{2} \phi^2 \quad (3.34)$$

tulee täyttää differentiaaliyhtälö (3.33). Hyödyntämällä yhtälöitä (3.33) ja (3.34) saadaan toiselle momentille m^ϕ differentiaaliyhtälö

$$\nabla \cdot (\rho \mathbf{u} m^\phi) = \nabla \cdot (\Gamma \phi \nabla m^\phi) - \Gamma \phi (\nabla \phi \cdot \nabla \phi) + Q \phi \phi, \quad (3.35)$$

jolle yleensä saadaan nolosta poikkeava tulos, koska yhtälön (3.33) numeerinen ratkaisu ei ole täysin tarkka. Tästä syystä toiselle momentille formuloitua differentiaaliyhtälöä (3.35) voidaan käyttää virhe-estimaattina, kun se ratkaistaan jokaista kontrollitulavuutta kohden. Integroimalla jokaisen kontrollitulavuuden yli saadaan

$$\bar{r}_{m,c}(m^\phi) = \int_{V_c} [\nabla \cdot (\rho \mathbf{u} m^\phi) - \nabla \cdot (\Gamma \phi \nabla m^\phi) - Q \phi \phi + \Gamma \phi (\nabla \phi \cdot \nabla \phi)] dV, \quad (3.36)$$

jossa tarvittavat m^ϕ -arvot tunnetaan entuudestaan ja muut termit saadaan muodostettua tunnetusta numeerisesta ratkaisusta siinä käytettyjen diskretisointimenetelmien avulla. Konvektiotermin muodostamisessa suositellaan käytettäväksi keskeisdifferenssimenetelmää, jotta tarvittavat rajapintojen arvot interpoloidaan toisen kertaluvun menetelmällä. Momenttimenetelmä voidaan myös yleistää vektorimuuttujille. (Jasak 1996, s. 168–170)

Vaikeutena yhtälön (3.36) käyttämisessä on kuitenkin se, että ratkaisun yksikkö ei vastaa alkuperäisen muuttujan ϕ yksikköä, vaan lopputuloksen yksikkö on aina muotoa

$[\phi]^2 [L]^3 / [T]$. Tästä syystä vaaditaan keino normalisoida momenttivilhe-estimaatin tulokset. Ajatellaan muuttujan ϕ efektiivisen nopeusskaalan riippuvan sen konvektiivisesta ja diffuusiivisesta siirtonopeudesta. Tällöin yksiköiden nojalla pätee

$$u_{eff} = |\mathbf{u}| + \frac{\Gamma\phi}{\rho h}, \quad (3.37)$$

jossa h riippuu lokaalista kontrollitulavuuden koosta. Nopeusskaalan u_{eff} avulla voidaan määrätä myös ominaisaikaskaala muodossa

$$T_{eff} = \frac{h}{u_{eff}} = \frac{h^2}{|\mathbf{u}|h + \frac{\Gamma\phi}{\rho}}, \quad (3.38)$$

jonka avulla voidaan normalisoida kontrollitulavuuskohtainen momenttijäännös $\bar{r}_{m,c}(m^\phi)$, kun jaetaan tämä vielä kontrollitulavuuden tilavuudella V_c . Näin ollen lokaalina momenttivilhe-estimaatina $\eta_{m,c}$ voidaan käyttää muotoa

$$\eta_{m,c}(\phi) = 2 \sqrt{\frac{|\bar{r}_{m,c}(m^\phi)| T_{eff}}{V_c}}, \quad (3.39)$$

joka perustuu neliölliseen riippuvuuteen (3.34). Virhe-estimaatin yksikkö vastaa nyt mielivaltaista ϕ -muuttujaa, eli johdettu virhe-estimaatti kuvaa lokaalia virhettä. (Jasak 1996, s. 170–171)

Momenttivilhe-estimaatin voidaan osoittaa täyttävän virhe-estimaateilta vaaditun asymptotisuusominaisuuden, jos yhtälö (3.36) diskretisoidaan neljännen kertaluvun menetelmillä. Syynä tähän diskretisointivaatimukseen on se, että toisen momentin m^ϕ variaatio on neliöllinen eikä lineaarinen. Käytännössä yleensä tyydytään toisen kertaluvun diskretisointimenetelmiin, koska korkeamman asteen menetelmien tuoma vaikutus momenttijäännöksen (3.36) on varsin pieni. Yksinkertaistuksella on kuitenkin se haittapuoli, että liian harvoilla laskentaverkoilla neljännen kertaluvun termien suhde toisen kertaluvun termeihin nähden saattaa olla liian suuri, jolloin myös estimaatin (3.39) suorituskyky kärsii. Tästä huolimatta momenttivilhe-estimaatti kuvaa diskretisointivirhettä huomattavasti paremmin kuin esimerkiksi Taylorin sarjakehitelmiin pohjautuvat estimaatit. Menetelmän heikkoutena voidaan pitää sitä, että se ei välttämättä kuvaa diskretisointivirhettä täysin oikein edellä mainitusta neljännen kertaluvun diskretisointivaatimuksesta johtuen, kun laskentaverkko tihenee. (Jasak 1996, s. 171–172)

Momenttivilhe-estimaatti on selkeästi eksplisiittinen virhe-estimaatti, sillä diskretisointivirheestä saadaan arvio jo olemassa olevan numeerisen ratkaisun perusteella, ja neljännen kertaluvun diskretisointimenetelmien ansiosta myös asymptotisuusehto täyttyy. Tämän

lisäksi Momenttivirhe-estimaatti vaikuttaa olevan siinä mielessä perinteinen eksplisiittinen virhe-estimaatti, että sen avulla ei saatu johdettua tarkkaa ylärajaa virheelle. Voidaan olettaa, että momenttijäännöksen $\bar{r}_{m,C}$ avulla muodostetuilla momenttivirhe-estimaateilla $\eta_{m,C}$ ei voida suoraan approksimoida diskretisointivirheen energianormia, vaikka Jasak (1996) ei tätä suoranaisesti tuokaan esille. Momenttimenetelmä pikemminkin antaa vain lokaalisti adaptiivisuuden kannalta käyttökelpoista tietoa diskretisointivirheestä.

Globaalin virhearvion muodostamisen kannalta toisena heikkoutena momenttimenetelmässä on se, että se ei ota huomioon vuojäännöksiä. Toisaalta momenttijäännöksen yksikkö ei myöskään vastaa alkuperäisen yhtälön yksikköä, mikä estää suoraan eksplisiittisen virhe-estimaatin (3.12) kaltaisen virhe-estimaatin muodostamisen. Kyseisen estimaatin muodostaminen olisi siinä mielessä tärkeää, että tämän avulla saataisiin myös globaali virhearvio summaamalla lokaalit virhe-estimaatit keskenään. Onneksi kontrollitulavuuskohtaisten jäännösten muodostamiseen löytyy kuitenkin parempia menetelmiä, kuten luvussa 3.4 osoittautuu. Työssä sovelletaan momenttivirhe-estimaattia sellaisenaan, koska menetelmän käyttö on varsin suoraviivaista.

3.4 Jäännösmenetelmä

Kontrollitulavuusmenetelmässä jäännösten muodostaminen on elementtimenetelmään nähden hankalampaa, sillä kontrollitulavuuksien keskipisteiden arvot periaatteessa toteutuvat ratkaistavat differentiaaliyhtälöt ratkaisijalle asetetun toleranssin tarkkuudella. Ratkaisusta saatavat rajapintojen arvot saattavat kuitenkin poiketa lineaarisella interpoloinnilla saatavista arvoista laskentaverkon muotovirheiden tai käytettyjen diskretisointimenetelmien vuoksi. Tällöin kontrollitulavuuksissa saattaa esiintyä diskretisointivirhettä jäännösten muodossa, sillä virtauskenttä ei muutu lineaarisesti kontrollitulavuuksien välillä. Jäännösten muodostamisessa täytyy käyttää siis hieman kekseliäisyyttä. Kontrollitulavuuksien sisällä mielivaltaisen pisteen A arvo voidaan lineaarisen variaation nojalla määrittää kontrollitulavuuden keskipisteessä tunnetun gradientin $\nabla\phi_C$ avulla yhteydestä

$$\phi_A = \phi_C + (\nabla\phi)_C \cdot (\mathbf{x}_A - \mathbf{x}_C), \quad (3.40)$$

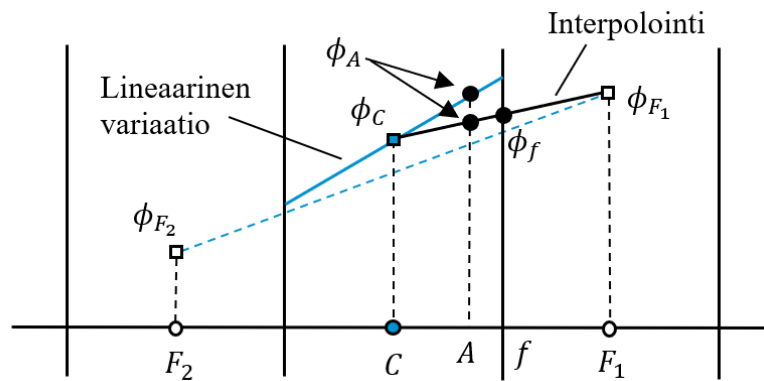
jossa ϕ_C ja $\nabla\phi_C$ ovat molemmat tunnettuja. Toisaalta kahden kontrollitulavuuden välillä voidaan käyttää myös etäisyyteen pohjautuvaa interpolointia

$$\phi_A = g_C\phi_C + (1 - g_C)\phi_F, \quad (3.41)$$

missä interpolaatiokertoimelle pätee yhteys

$$g_C = \frac{\|\mathbf{x}_F - \mathbf{x}_A\|}{\|\mathbf{x}_A - \mathbf{x}_C\|} = \frac{d_{FA}}{d_{FC}}, \quad (3.42)$$

mutta yhtälöstä (3.41) interpoloimalla saatu ϕ_A -arvo ei välttämättä aina vastaa gradientilla määritettyä arvoa (3.40). Interpoloidut arvot ovat yhtäsuuret vain siinä tapauksessa, jos numeerinen ratkaisu vastaa tarkkaa ratkaisua. Mielenkiintoinen havainto saadaan, jos piste A sijoitetaan kahden kontrollitilavuuden välisen rajapinnan pintakeskiöön, jolloin muodostuu jopa kolme eri arvoa käyttämällä yhteyksiä (3.40) ja (3.41). Ensimmäinen arvo saadaan interpoloimalla suoraan yhtälöstä (3.40) ja toinen interpoloimalla sama piste naapurikontrollitilavuuden arvoilla ϕ_F , $\nabla\phi_F$ ja $\mathbf{x}_A - \mathbf{x}_F$. Kolmas arvo saadaan etäisyyteen pohjautuvalla interpoloinnilla (3.41). Jos arvoissa on eroavaisuuksia, näiden avulla voidaan rakentaa jokaiselle kontrollitilavuudelle numeerisen ratkaisun jäännös. (Jasak 1996, s. 173–176) Jäännöstä on havainnollistettu kuvassa 3.3.



Kuva 3.3 Kontrollitilavuuksien välillä esiintyvä jäännös. Mukailtu lähteestä (Jasak 1996, s. 174).

Pelkästään interpoloitujen arvojen perusteella muodostettu jäännös ei varsinaisesti ota differentiaaliyhtälön konvektio- tai diffuusio-osuutta huomioon. Esimerkiksi yhtälön (3.33) tapauksessa saadaan muodostettua edellä mainitut seikat huomioiva virhe-estimaatti, jos jäännös määritetään jokaiselle kontrollitilavuudelle integraalista

$$\bar{r}_{r,C}(\phi) = \int_{V_C} [\nabla \cdot (\rho \mathbf{u} \phi) - \nabla \cdot (\Gamma^\phi \nabla \phi) - Q^\phi] dV = \sum_f^{n(f)} [(\rho \mathbf{u} \phi)_f - (\Gamma^\phi \nabla \phi)_f] - Q^\phi V_C, \quad (3.43)$$

jossa rajapintojen arvot saadaan lineaarisen interpolaation

$$\phi_f = \phi_C + (\nabla \phi)_C \cdot (\mathbf{x}_f - \mathbf{x}_C), \quad (3.44)$$

avulla ja rajapinnan pintakeskiössä oleva gradientti oletetaan samaksi kuin kontrollitilavuuden keskipisteen gradientti, eli

$$(\nabla \phi)_f = (\nabla \phi)_C. \quad (3.45)$$

Hienoutena jäännöksessä (3.43) on se, että sillä saadaan esille myös diskretisointivirheet, jotka aiheutuvat laskentaverkon muotovirheistä, kuten esimerkiksi vinoumasta, koska ϕ_f määritetään todellisessa pintakeskiössä f , eikä leikkauspisteessä f' . Myös diskretisointivirheiden kannalta kriittinen numeerinen diffuusio vaikuttaa jäännöksen suuruuteen, koska se lisää konvektiojäännöstä suoraan. Merkittävänä seikkana edellä muodostetussa jäännöksessä on se, ettei se periaatteessa rajoita numeerisessa ratkaisussa käytettäviä diskretisointimenetelmiä lainkaan, jos pidetään huoli siitä, että muuttujan ϕ variaatio valitaan asianmukaisesti diskretisointimenetelmien kertaluvun perusteella. Ei myöskään ole syytä olettaa, että menetelmän suorituskyky olisi suoranaisesti riippuvainen laskentaverkon koosta. Voidaan myös todistaa, että menetelmä suppenee asymptoottisesti kohti todellista diskretisointivirhettä. Käyttökelpoisen virhe-estimaatin muodostaminen vaatii vielä jäännösten normalisoinnin, koska niiden yksikkö on $[\phi][L]^3/[T]$. (Jasak 1996, s. 176–179)

Jäännösvirhe (3.43) kannattaa normalisoida momenttivirhemenetelmän tavoin yksittäisten kontrollitilavuuksien ominaisuuksien avulla. Jäännösvirheessä esiintyy konvektio-, diffuusio- ja lähdetermi, joten on etsittävä keino normalisoida näiden termien yhteisvaikutus. Diffuusiotermin tapauksessa on järkevintä muodostaa kontrollitilavuuskohtainen diffuusiovuon tilavuuskeskiarvo

$$F^{\phi,D} = \frac{1}{V_C} \sum_f^{n(f)} \left[\frac{S_f(\Gamma\phi)_f}{d_{CF}} \right]. \quad (3.46)$$

Vastaavasti samanlaista tilavuuskeskiarvoa voidaan hyödyntää myös konvektion tapauksessa, kuitenkin sillä poikkeuksella, että huomioidaan vain kontrollitilavuudesta poispäin virtaavat vuot. Konvektiotermi otetaan täten huomioon keskimääräisenä konvektiovuona

$$F^{\phi,C} = \frac{1}{V_C} \sum_f^{n(f)} \max(\dot{m}_f, 0), \quad (3.47)$$

mutta normalisointi ei ole vielä täysin valmis. Tarvitaan nimittäin keino normalisoida lähdetermien vaikutus, sillä ratkaistava yhtälö saattaa sisältää vuomaisia implisiittisiä lähdetermejä, jotka riippuvat ϕ_C -arvoista, kuten esimerkiksi nopeuskentän divergenssi. Tällöin niiden vaikutus täytyy ottaa huomioon lisäämällä ne erilliseen $F^{\phi,Q}$ -termiin. Sen sijaan eksplisiittisillä lähdetermeillä ei ole vaikutusta normalisointiin, koska nämä oletetaan tarkoiksi, eikä näillä ole edellä mainittua ϕ_C riippuvuutta. Ottamalla kaikki edellä esitetyt normalisointikertoimet huomioon, saadaan normalisointikerroin

$$F^\phi = F^{\phi,D} + F^{\phi,C} + F^{\phi,Q}, \quad (3.48)$$

jota voidaan soveltaa kontrollitilavuuskohtaiseen jäännökseen (3.43). Lopulta saadaan muodostettua lokaali jäännösvirhe-estimaatti

$$\eta_{r,c}(\phi) = \frac{\bar{r}_{r,c}(\phi)}{V_C F \phi, c}, \quad (3.49)$$

jonka yksikkö on sama kuin mielivaltaisen ϕ -muuttujan. Vektorimuuttujan tapauksessa jäännösvirhe-estimaatti tuottaa vektorimuotoisen virhe-estimaatin, josta saadaan kätevästi myös skalaarivirhe laskemalla kyseisen virhe-estimaatin normi. (Jasak 1996, s. 179–181) Implisiittisen lähdetermin lisääminen normalisointikertoimeen (3.48) on varsin selkeä, jos tarkastellaan aiemmin luvussa 2.7 johdettua lähdetermin diskretisointia. Lähdeterminä voidaan tunnistaa vuomainen ϕ_C -muuttujasta riippuva termi ja toinen täysin eksplisiittisesti arvioitava termi.

Jäännösmenetelmällä saadulla jäännöksellä voitaisiin periaatteessa johtaa elementtimenetelmän kaltainen eksplisiittinen virhe-estimaatti (3.12), koska löydettiin keino saada esille kontrollitilavuuskohtainen jäännös (3.43). Tosiasiassa kyseisen virhe-estimaatin muodostaminen saatiin taidokkaasti kierrettyä, koska jäännös ottaa huomioon gradienttien hyppäykset ja toimii itsessään jo hyvänä virhe-estimaattina, kun se vielä normalisoidaan. Sen sijaan vaihtokaupassa menetettiin mahdollisuus approksimoida diskretisointivirheen globaalia energianormia, koska virhe-estimaatti (3.49) ei selkeästi ole virhe-estimaattia (3.12) vastaava. Ei ole selvää, voidaanko jäännösvirhe-estimaatit summata ja täten arvioida numeerisen ratkaisun virheen ylärajaa. Joka tapauksessa tämä riippuu vielä tuntemattomasta vakiosta, joten ongelma ei ole välttämättä vakava, koska kontrollitilavuusmenetelmään on johdettu myös luotettavan ylärajan antava implisiittinen virhe-estimaatti, jota käsitellään luvussa 3.5. Jäännösvirhe-estimaatti on ehdottomasti hyvin diskretisointivirhettä havainnollistava, koska tämän yksikkö vastaa ϕ -muuttujan yksikköä. Edellä mainituista syistä jäännösvirhe-estimaattia sovelletaan tässä työssä sellaisenaan.

3.5 Tasapainotettujen jäännösten menetelmä

Tarkastellaan aiemmin luvussa 3.1 esillä ollutta malliongelmaa (3.5) ottamalla huomioon mielivaltaisen diffuusiokerroin a , jolloin saadaan ongelma

$$-\nabla \cdot (a \nabla u) + cu = \hat{f}, \quad (3.50)$$

ja tälle asetetaan oleelliset ja luonnolliset reunaehdot

$$\begin{cases} u = 0, & \text{reunalla } \Gamma_D, \\ a \nabla u \cdot \mathbf{n}_f = \hat{g}, & \text{reunalla } \Gamma_N, \end{cases} \quad (3.51)$$

jossa \mathbf{n}_f on pinnan normaalin suuntainen yksikkövektori. Malliongelmalle (3.50) tunnetaan diskretisointivirheen globaali energianormi muodossa

$$\|e\|_E = \int_{\Omega} (a \nabla e \cdot \nabla e + ce^2) dV, \quad (3.52)$$

jonka kontrollitilavuuskohtaista osuutta $\|e\|_{E,C}$ yritetään tasapainotettujen jäännösten menetelmässä approksimoida jollakin implisiittisellä virhe-estimaatilla φ_C . Ratkaisemalla jokaisessa kontrollitilavuudessa lokaali ongelma

$$-\nabla \cdot \nabla \xi_C = r_C \quad (3.53)$$

reunaehdoin

$$\begin{cases} \xi_C = 0, & \text{reunalla } \partial V_C \cap \Gamma_D, \\ \nabla \xi_C \cdot \mathbf{n}_f = R_C, & \text{reunalla } \partial V_C \setminus \Gamma_D, \end{cases} \quad (3.54)$$

saadaan aikaan jokaista kontrollitilavuutta kohden, gradientista $\nabla \xi_C$ ja kertoimesta a riipuva, implisiittinen virhe-estimaatti

$$\|\varphi(\nabla \xi)\|_{E,C}^2 = \int_{V_C} \frac{1}{a} \nabla \xi_C \cdot \nabla \xi_C dV. \quad (3.55)$$

Virhe-estimaatti (3.55) on vastine todelliselle energianormille $\|e\|_{E,C}$. Mikäli kyseinen implisiittinen virhe-estimaatti tunnetaan jokaisessa kontrollitilavuudessa, voidaan lokaalit virhe-estimaatit summata globaalisti, eli tällöin saadaan

$$\|e\|_E^2 \leq \sum_{C=1}^{n(C)} \|\varphi(\nabla \xi)\|_{E,C}^2, \quad (3.56)$$

joka antaa diskretisointivirheen energianormille luotettavan ylärajan. (Jasak 1996, s. 181–183) Implisiittisen jälkikäteisvirhearvion muodostaminen on siis selkeästi työläämpää kuin aiemmin esitettyjen momenttinvirhe- tai jäännösvirhe-estimaattien käyttö, koska estimaatti (3.55) vaatii vielä erillisen ongelman ratkaisemisen. Sen sijaan lisääntynyt työmäärä on hyvin perusteltavissa, sillä tarkka yläraja (3.56) ei enää riipu tuntemattomasta kertoimesta C_0 . Lokaalien ongelmien (3.53) reunaehdot kuitenkin vaativat vielä lisähuomiota, ennen kuin ongelma voidaan ratkaista, joten näihin keskitytään seuraavaksi.

Reunaehdoissa esiintyvä vuojäännös R_C vaatii käsittelyä, ennen kuin reunaehdoja (3.54) voidaan soveltaa lokaaliin ongelmaan (3.53). Sen sijaan jäännöksen r_C tilavuusintegraali tunnetaan, koska tämä saadaan yhteydestä

$$\int_{V_C} r_C(u) dV = -\bar{r}_C(u), \quad (3.57)$$

mikä vastaa luvussa 3.4 esitettyä jäännös menetelmää. Määritetään reunaehdoissa (3.54) esiintyvä vuojäännös R_C alkuperäisen ongelman reunaehdoista riippuen kahteen osaan

$$R_C = \begin{cases} R_C = \hat{g} - a \nabla u_h \cdot \mathbf{n}_f, & \text{reunalla } \partial V_C \cap \Gamma_N, \\ R_C = -\alpha_C \llbracket a \nabla u_h \cdot \mathbf{n}_f \rrbracket, & \text{reunalla } \partial V_C \setminus \Gamma_N, \end{cases} \quad (3.58)$$

missä $[\![\nabla u_h \cdot \mathbf{n}_f]\!]_C$ kuvaa vuohyppäystä kontrollitilavuuksien välisillä rajapinnoilla. Kyseisille rajapinnoille ei ole määrätty luonnollisia reunaehtoja alkuperäisessä ongelmassa, ja α_C on jokin tuntematon kerroin. Rajapinnoilla tapahtuva vuohyppäys saadaan määritettyä gradienttien avulla yhteydestä

$$[\![a\nabla u_h \cdot \mathbf{n}_f]\!] = [a(\nabla u_h)_f \cdot \mathbf{n}_f]_C - [a(\nabla u_h)_f \cdot \mathbf{n}_f]_F. \quad (3.59)$$

Gradienttihyppäysten lisäksi numeerisessa ratkaisussa saattaa kuitenkin esiintyä myös muuttujan u hyppäyksiä, eli

$$[\![u]\!] = [(u_h)_f]_C - [(u_h)_f]_F. \quad (3.60)$$

Ennen kontrollitilavuuksien välisissä vuojäännöksissä esiintyvän α_C -kertoimen tarkempaa määrittelyä, tarvitaan vielä keino interpoloida vuo $a\nabla u_h \cdot \mathbf{n}_f$ kontrollitilavuuden sisällä. Kyseinen vuo voidaan interpoloida yhteydestä

$$\langle a\nabla u_h \cdot \mathbf{n}_f \rangle_\alpha = \alpha_C a[(\nabla u_h)_f]_C \cdot \mathbf{n}_f + (1 - \alpha_C) a[(\nabla u_h)_f]_F \cdot \mathbf{n}_f \quad (3.61)$$

ja samalla tavalla voidaan myös interpoloida muuttuja u

$$\langle u_h \rangle_\alpha = \alpha_C [(u_h)_f]_C + (1 - \alpha_C) [(u_h)_f]_F. \quad (3.62)$$

Tuntemattomien α_C -kertoimien määrittäminen monimutkaistaa implisiittisen virhe-estimaatin muodostamista siten, että yhteyksissä (3.61) ja (3.62) esiintyvät kertoimet täytyvät olla samat kuin vuojäännöksissä (3.58). Tasapainotettujen jäännösten menetelmässä ideana on määrittää α_C -kerroin sillä tavalla, että tämän avulla muodostetuilla reunaehdoilla saadaan merkityksellinen ratkaisu yhtälöstä (3.53). (Jasak 1996, s. 183–184)

Koska α_C -kerroin vaikuttaa suoraan lokaalin ongelman reunaehtoihin, tulee kertoimien avulla määritettyjen vuojäännöksien olla tasapainossa, eli jokaisessa kontrollitilavuudessa tulee toteutua ehto

$$\int_{V_C} r_C dV + \oint_{\partial V_C} R_C dA = 0, \quad (3.63)$$

joka varmistaa sen, että ongelmalle (3.53) saadaan ratkaisu. Tase-ehdon (3.63) varmistaminen hankaloittaa menetelmän käyttöä entuudestaan. Osoittautuu, että jakamalla tase-ehto malliongelman oleellisten ja luonnollisten reunaehtojen avulla pienempiin palasiin, ja hyödyntämällä vuojäännöksiä (3.58), saadaan

$$\int_{V_C} [\hat{f} - c(u_h)_C] dV + \oint_{\partial V_C \cap \Gamma_N} \hat{g} dA + \oint_{\partial V_C \setminus \Gamma_N \setminus \Gamma_D} \langle a\nabla u_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)} dA = 0, \quad (3.64)$$

jossa esiintyvä interpoloitu vuo $\langle a\nabla u_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)}$ on jo yhteydestä (3.61) tuttu. Hankaluu-
tena on se, miten vuon interpolointi tapahtuu, jotta ehto (3.64) täyttyy ja samalla myös
alkuperäinen ongelma (3.50) toteutuu. Tase-ehtoa voidaan helpoiten hyödyntää, kun lau-
sutaan gradienttihyppäyksiä sisältävät vuojäännökset sellaisessa muodossa, että niissä
esiintyy tämä kyseinen vuo $\langle a\nabla u_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)}$. Muokkaamalla kontrollitilavuuksien raja-
pintojen välillä olevaa vuojäännöstä muotoon

$$\begin{aligned} -\alpha_C \llbracket a\nabla u_h \cdot \mathbf{n}_f \rrbracket &= -\alpha_C [a(\nabla u_h)_C \cdot \mathbf{n}_f - a(\nabla u_h)_F \cdot \mathbf{n}_f] \\ &= (1 - \alpha_C)a(\nabla u_h)_C \cdot \mathbf{n}_f + \alpha_C a(\nabla u_h)_F \cdot \mathbf{n}_f - a(\nabla u_h)_C \cdot \mathbf{n}_f \\ &= \langle a\nabla u_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)} - a(\nabla u_h)_C \cdot \mathbf{n}_f, \end{aligned} \quad (3.65)$$

saadaan siihen aikaiseksi haluttu suure $\langle a\nabla u_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)}$. Kyseinen interpoloitu vuo on
kuitenkin luonnostaan saatavilla kontrollitilavuusmenetelmän tuloksista. Osoittautuu,
että α_C -kertoimia ei tosiasiaassa tarvitse määrittää, koska tunnetaan yhteys

$$\langle a\nabla u_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)} = a \frac{(u_h)_F - (u_h)_C}{d_{CF}}. \quad (3.66)$$

Kyseiseen interpoloituun suureeseen voidaan lisätä myös epäortogonaalisuuden aiheut-
tama lisäkorjaus, jos laskentaverkossa esiintyy epäortogonaalisuutta. (Jasak 1996, s. 184–
187) Edellä saatiin varmistettua, että lokaali ongelma voidaan ratkaista jokaisessa kont-
rollitilavuudessa, mistä ei ole takeita, jos tase-ehtoa ei täytetä.

Lokaaleille ongelmille saadaan täten tase-ehdon täyttävät reunaehdot

$$R_C = \begin{cases} R_C = \hat{g} - a\nabla u_h \cdot \mathbf{n}_f, & \text{reunalla } \partial V_C \cap \Gamma_N \\ R_C = -a \frac{(u_h)_F - (u_h)_C}{d_{CF}} + a(\nabla u_h)_C \cdot \mathbf{n}_f. & \text{reunalla } \partial V_C \setminus \Gamma_N \end{cases} \quad (3.67)$$

Menetelmä voidaan yleistää myös mielivaltaisen skalaarimuuttujan säilymisyhtälölle,
kun otetaan huomioon myös konvektiotermi. Tällöin reunaehdoissa täytyy esiintyä myös
muuttujan u hyppäykset, mutta periaate on sama kuin malliongelman tapauksessa ja lop-
putuloksena saadaan estimaatin (3.55) kaltainen implisiittinen virhe-estimaatti jokaiselle
kontrollitilavuudelle. (Jasak 1996, s. 186–189) Skalaarimuuttujalle yleistystä ei kuiten-
kaan esitetä, vaan siirrytään suoraan Navierin–Stokesin yhtälöiden implisiittiseen virhe-
estimaattiin, joka ei loppujen lopuksi eroa kovinkaan paljoa malliongelman tapauksesta.

Navierin–Stokesin yhtälöiden tapauksessa numeerisen ratkaisun diskretisointivirhe riip-
puu nopeusvektorin diskretisointivirheestä

$$\mathbf{e}_u = \mathbf{u} - \mathbf{u}_h \quad (3.68)$$

ja myös paineen diskretisointivirheestä

$$E = p - p_h, \quad (3.69)$$

koska Navierin–Stokesin yhtälöt ovat kytkettyneitä toisiinsa painegradientin kautta. Paineen diskretisoinnissa tapahtuneet virheet lisäävät koko numeerisen ratkaisun virhettä aiheuttamalla nopeuskenttään divergenssiä. Tästä syystä energianormin sijaan muodostetaan tätä vastaava normi

$$\|(\mathbf{e}_u, E)\|_*^2 = \int_{\Omega} \nu(\nabla \mathbf{e}_u : \nabla \mathbf{e}_u) dV + \int_{\Omega} E^2 dV, \quad (3.70)$$

mitä approksimoidaan implisiittisellä virhe-estimaatilla φ_C . (Jasak 1996, s. 189–192)

Tasapainotettujen jäännösten menetelmän soveltaminen Navierin–Stokesin yhtälöihin vaatii monimutkaisempien reunaehtojen asettamisen lokaalille ongelmalle. Navierin–Stokesin yhtälöiden virhe-estimaatti saadaan, kun ratkaistaan vektorimuotoinen lokaali ongelma

$$-\nabla \cdot \nabla \boldsymbol{\xi}_C = \mathbf{r}_C \quad (3.71)$$

reunaehdoin

$$\begin{cases} \nabla \boldsymbol{\xi}_C \cdot \mathbf{n}_f = (\mathbf{R}_v)_C, & \boldsymbol{\xi}_C = 0, & \text{reunalla } \partial V_C \cap \Gamma_D, \\ \nabla \boldsymbol{\xi}_C \cdot \mathbf{n}_f = \mathbf{R}_C, & & \text{reunalla } \partial V_C \setminus \Gamma_D, \end{cases} \quad (3.72)$$

missä esiintyvä jäännösvektori \mathbf{r}_C saadaan tutulla tavalla yhteydestä

$$\mathbf{r}_C = \int_{V_C} [\mathbf{f} - \nabla p_h - \nabla \cdot (\rho \mathbf{u}_h \mathbf{u}_h) + \nabla \cdot (\nu \nabla \mathbf{u}_h)] dV, \quad (3.73)$$

jossa \mathbf{f} sisältää Navierin–Stokesin yhtälöiden lähdetermin. Tase-ehdon täyttävien vuojäännösten avulla saadaan kontrollitulavuuksien välisille rajapinnoille asetettua reunaehdot

$$\mathbf{R}_C = \dot{m}_f [-\langle \mathbf{u}_h \rangle_{(1-\alpha)} + (\mathbf{u}_h)_f] - S_f [-\langle \nu \nabla \mathbf{u}_h \cdot \mathbf{n}_f \rangle_{(1-\alpha)} + \nu (\nabla \mathbf{u}_h)_f \cdot \mathbf{n}_f] \quad (3.74)$$

ja pinnoille, joille on asetettu nopeudelle nollagradientti reunaehto Navierin–Stokesin yhtälöissä, käytetään yksinkertaistettua muotoa

$$\mathbf{R}_C = \dot{m}_f [-\langle \mathbf{u}_h \rangle_{(1-\alpha)} + (\mathbf{u}_h)_f] - S_f \nu (\nabla \mathbf{u}_h)_f \cdot \mathbf{n}_f. \quad (3.75)$$

Vastaavasti pinnoille, joille on asetettu Navierin–Stokesin yhtälöissä oleelliset reunaehdot, saadaan edelleen yksinkertaisempi reunaehto

$$(\mathbf{R}_v)_C = \dot{m}_f [-\langle \mathbf{u}_h \rangle_{(1-\alpha)} + (\mathbf{u}_h)_f], \quad (3.76)$$

joka ei luonnostaan riipu gradienttien suuruudesta, koska luonnollisia reunaehtoja ei ole annettu näille alueille, eikä siten niissä esiinny hyppäyksiä. Kaikissa reunaehdoissa interpoloitu arvo $\langle \mathbf{u}_h \rangle_{(1-\alpha)}$ määritetään samalla tavalla kuin konvektiotermin diskretisoinnissa. (Jasak 1996, s. 191–192)

Ratkaisemalla jokaisessa kontrollitilavuudessa yhtälö (3.71) saadaan lokaali implisiittinen virhe-estimaatti

$$\|\varphi(\nabla \xi, \nabla \cdot \mathbf{u}_h)\|_C^2 = \int_{V_C} \frac{1}{\nu} \nabla \xi_C : \nabla \xi_C \, dV + \int_{V_C} [\nabla \cdot (\mathbf{u}_h)_C]^2 \, dV \quad (3.77)$$

ja tämän avulla globaaliksi virhearvioksi saadaan

$$\|(\mathbf{e}_u, E)\|_*^2 \leq \sum_{C=1}^{n(C)} \|\varphi(\nabla \xi, \nabla \cdot \mathbf{u}_h)\|_C^2, \quad (3.78)$$

joka kuvaa Navierin–Stokesin yhtälöiden diskretisointivirheen energianormin ylärajaa luotettavasti. (Jasak 1996, s. 192)

Tarkempi johto tasapainotettujen jäännösten menetelmän soveltamisesta yksinkertaiseen malliongelmaan löytyy lähteestä (Ainsworth & Oden 1993) ja Navierin–Stokesin yhtälöille lähteestä (Oden et al. 1994). Menetelmää ei johdettu kyseisistä lähteistä siitä syystä, että kontrollitilavuusmenetelmän sovellus oli jo saatavilla.

Tarkastellaan vielä yksinkertaiselle malliyhtälölle johdettua lokaalia ongelmaa (3.53) ja sille asetettuja oleellisia ja luonnollisia reunaehtoja (3.54). Jasakin (1996, s. 192–193) mukaan ongelma on hyvin aseteltu ja suoraan lokaalisti ratkaistavissa, jos kontrollitilavuuden pinnalle on asetettu ainakin yksi oleellinen reunaehto, mutta määrittelemätön, jos pinnoille on asetettu vain luonnollisia reunaehtoja. Seuraavaksi esitetään, miten lokaali ongelma voidaan ratkaista edellä mainitun havainnon avulla ilman globaalien yhtälöryhmän muodostamista.

Implisiittisessä virhe-estimaatissa (3.55) esiintyvä $\nabla \xi_C$ -gradientti voidaan ratkaista jokaisessa kontrollitilavuudessa, jos oletetaan, että ξ muuttuu lineaarisesti jokaisen kontrollitilavuuden sisällä. Muuttujan todellista arvoa ei välttämättä tarvita, vaan virhe-estimaattiin muodostamiseen riittää tämän gradientti. Lineaarisen variaation nojalla kontrollitilavuuksien rajapinnoilla pätee

$$\xi_f = \xi_C + (\nabla \xi)_C \cdot (\mathbf{x}_f - \mathbf{x}_C) \quad (3.79)$$

ja tämän avulla voidaan koota kontrollitilavuuden keskipisteessä gradientti $\nabla \xi_C$ Greenin–Gaussin lauseen mukaisesti, eli yhteydestä

$$\nabla \xi_C = \frac{1}{V_C} \sum_f^{n(f)} \xi_f \mathbf{S}_f. \quad (3.80)$$

Tehdään myös toinen oletus, että keskimääräinen gradientti on vakio koko kontrollitilavuudessa, eli yhteys $\nabla \xi_C = \nabla \xi_f$ pätee. Edellä mainitun oletuksen sekä yhtälöiden (3.79) ja (3.80) avulla saadaan gradientti $\nabla \xi_C$ nyt yhteydestä

$$\nabla \xi_C = \frac{1}{V_C} \sum_f^{n(f)} [(\nabla \xi)_f \cdot (\mathbf{x}_f - \mathbf{x}_C)] \mathbf{S}_f, \quad (3.81)$$

jossa ξ_C -termin vaikutus jätettiin huomioimatta. Jos jätetään myös laskentaverkon epäortogonaalisuus huomioimatta, voidaan kontrollitilavuuden pinnoille asetettuja reunaehtoja $(R_C)_f = (\nabla \xi)_f \cdot \mathbf{n}_f$ soveltaa gradienttiin (3.81) yhteyden

$$(\nabla \xi)_f \cdot (\mathbf{x}_f - \mathbf{x}_C) = \|\mathbf{x}_f - \mathbf{x}_C\| (\nabla \xi)_f \cdot \mathbf{n}_f = d_{Cf} (\nabla \xi)_f \cdot \mathbf{n}_f \quad (3.82)$$

avulla. Gradientti saadaan ratkaistua edellä mainittujen reunaehtojen ansiosta yhteydestä

$$\nabla \xi_C = \frac{1}{V_C} \sum_f^{n(f)} d_{Cf} (\nabla \xi)_f \cdot \mathbf{n}_f \mathbf{S}_f = \frac{1}{V_C} \sum_f^{n(f)} d_{Cf} (R_C)_f \mathbf{S}_f, \quad (3.83)$$

mikäli kaikki vuojäännökset $(R_C)_f$ ovat tunnettuja. Tulos (3.83) pätee vain siinä tapauksessa, jos kontrollitilavuudessa ratkaistavalle lokaalille ongelmalle on asetettu pelkästään luonnollisia reunaehtoja. (Jasak 1996, s. 193–194)

Kaikki kontrollitilavuuden pinnat eivät kuitenkaan ole aina kahden kontrollitilavuuden välisiä rajapintoja, vaan osa pinnoista saattaa olla oleellisia reunaehtoja sisältäviä reunapintoja. Tällöin gradientin $\nabla \xi_C$ laskentaa joudutaan hieman monimutkaistamaan. Lokaali ongelma (3.53) voidaan diskretisoida kontrollitilavuusmenetelmällä muotoon

$$- \sum_f^{n(f)} (\nabla \xi_C \cdot \mathbf{S})_f = r_C V_C, \quad (3.84)$$

jossa esiintyvä summatermi voidaan jakaa kahtia reunaehtojen perusteella, eli

$$\sum_f^{n(f)} (\nabla \xi_C \cdot \mathbf{S})_f = \sum_g^{n(g)} (\nabla \xi_C \cdot \mathbf{S})_g + \sum_v^{n(v)} (\nabla \xi_C \cdot \mathbf{S})_v, \quad (3.85)$$

jossa alaindeksi g viittaa luonnollisia reunaehtoja käyttäviin pintoihin. Vastaavasti v viittaa pintoihin, joille on määrätty oleellinen reunaehto. Yhtälössä (3.85) olevat muuttujat tiedetään, sillä tunnetaan yhteys

$$(\nabla \xi_C)_g \cdot \mathbf{S}_g = S_g (\nabla \xi_C)_g \cdot \mathbf{n}_g = S_g (R_C)_g \quad (3.86)$$

kontrollitilavuudelle asetettujen luonnollisten reunaehtojen vuoksi ja vastaavasti oleellisten reunaehtojen avulla voidaan muodostaa yhteys

$$(\nabla \xi_C)_v \cdot \mathbf{S}_v = S_v (\nabla \xi_C)_v \cdot \mathbf{n}_v = S_v \frac{\xi_v - \xi_C}{d_{cv}}. \quad (3.87)$$

Soveltamalla vielä lokaalin ongelman oleellista reunaehto $\xi_v = 0$, saadaan yhtälöiden (3.84–3.87) nojalla ξ -muuttujan arvoksi määritettyä kontrollitilavuuden keskipisteessä

$$\xi_C = \frac{\int_{V_C} r_C dV + \sum_g^{n(g)} (R_C)_g S_g}{\sum_v^{n(v)} \frac{S_v}{d_{cv}}}. \quad (3.88)$$

Tuloksen (3.88) avulla voidaan vaihtaa oleelliset reunaehdot helpommin käytettäviksi luonnollisiksi reunaehdoiksi yhteydestä

$$(\nabla \xi_C)_v \cdot \mathbf{n}_v = \frac{-\xi_C}{d_{cv}}. \quad (3.89)$$

Näin kannattaa tehdä, koska sijoittamalla nämä muiden luonnollisten reunaehtojen kanssa yhtälöön (3.83), voidaan määrittää helposti $\nabla \xi_C$ -gradientti myös niille kontrollitilavuuksille, joissa yhdelle tai useammalle pinnalle on asetettu oleellinen reunaehto. Tulos (3.83) voidaan yleistää myös vektorimuotoiselle lokaalille ongelmalle. (Jasak 1996, s. 194–195)

Vaikka implisiittisen virhe-estimaatin muodostamisen teoria on huomattavan monimutkainen, saatiin käyttökelpoinen estimaatti, koska lokaalien ongelmien ratkaisu merkittävästi yksinkertaistuu. Hankaluus tasapainotettujen jäännösten menetelmässä kohdistuu enimmäkseen tarvittavien jäännösten muodostamiseen, mutta jäännösten muodostamisen jälkeen gradientin $\nabla \xi_C$ ratkaiseminen ei ole enää vaikea tehtävä.

Tasapainotettujen jäännösten menetelmän implisiittinen virhe-estimaatti on alun perin johdettu perinteisille Navierin–Stokesin yhtälöille. Jos oletetaan kaikki lähde-terminit eksakteiksi painegradientti pois lukien, voidaan menetelmä yleistää hyvin myös aiemmin esillä olleille RANS-yhtälöille. Tällöin lähde-terminiin \mathbf{f} sisällytetään esimerkiksi Reynoldsin jännitystensorin sisältämä termi. Myös momenttimenetelmän ja jäännös menetelmän tapauksessa näiden yleistys RANS-yhtälöille tapahtuu lähde-terminien muokkaamisella. Momenttimenetelmään ja jäännös menetelmään palataan tarkemmin neljännessä luvussa, jossa näitä sovelletaan.

4. ADAPTIIVISEN KONTROLLITILAVUUSMENETELMÄN OPENFOAM-TOTEUTUS

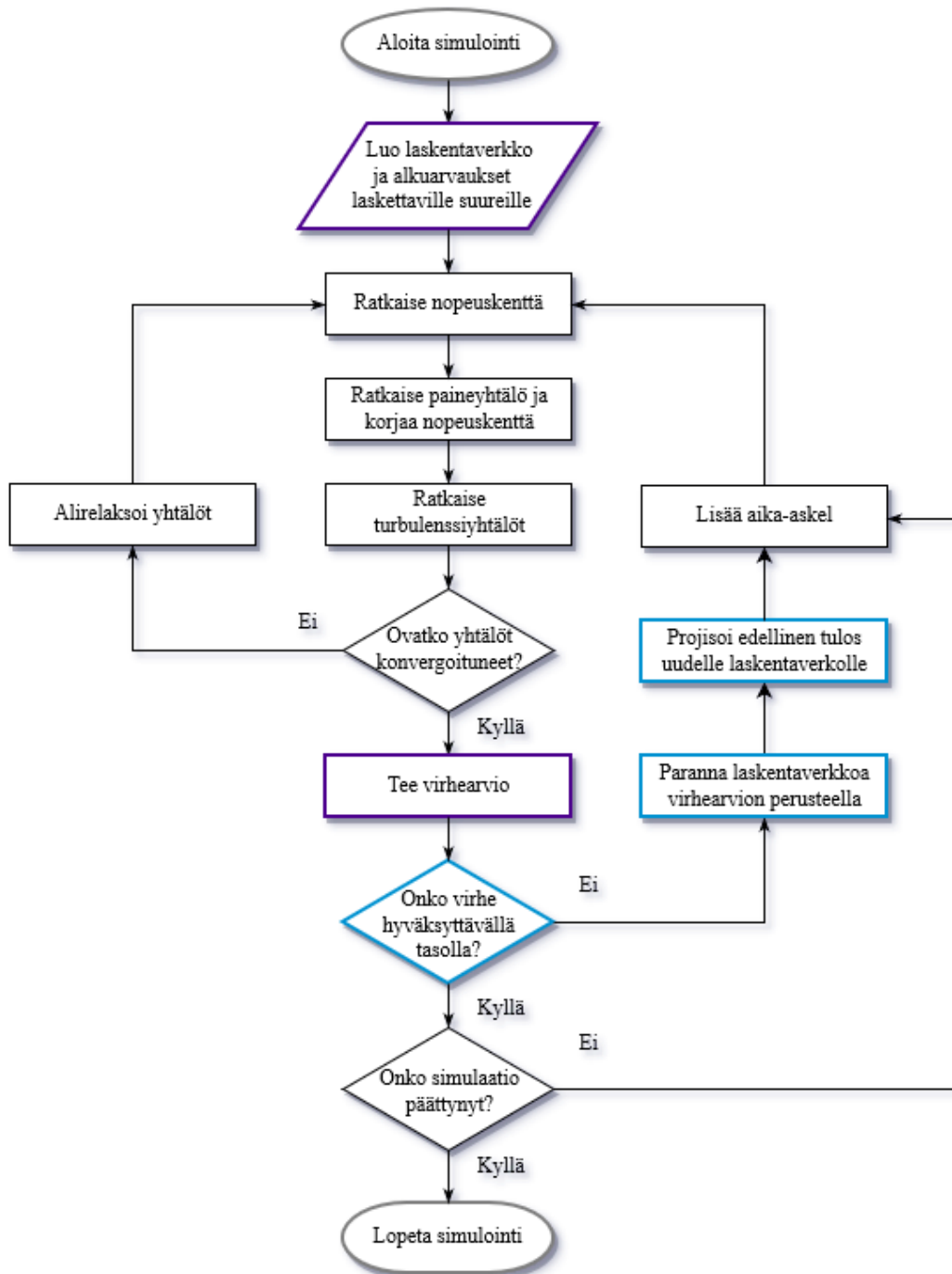
4.1 OpenFOAM-alustan valmiiden ominaisuuksien selvitys

Edellisissä luvuissa tarkasteltiin erilaisia diskretisointi- ja virhearviomenetelmiä sekä esitettiin myös työssä ratkaistavat yhtälöt. Seuraavaksi siirrytään soveltamaan teoriasta opittuja asioita ja algoritmien toimivuutta testataan käytännössä. Tätä varten OpenFOAM-toteutukseen on järkevintä valita ominaisuuksiltaan mahdollisimman valmis ratkaisija, johon voidaan toteuttaa helposti tarpeelliset adaptiivisuuden ja virhearvioihin liittyvät lisäominaisuudet, joiden toimivuutta työssä halutaan tutkia. OpenFOAM v5.0 -ohjelmiston käyttäjän käsikirjan (Greenshields 2017, s. 91–92) mukaan ohjelmistossa on suoraan saatavilla PIMPLE-algoritmiin pohjautuva *pimpleDyMFoam*-ratkaisija, jonka avulla on mahdollista ratkaista turbulentsia ja kokoonpuristumattomia virtauksia sekä muuttaa laskentaverkkoa sisäänrakennetuilla dynaamisilla ominaisuuksilla. Tätä ratkaisijaa kannattaa lähteä jatkojalostamaan, jotta työmäärä pysyy kohtuullisena.

PimpleDyMFoam-ratkaisijan käyttäminen adaptiivisen kontrollitulavuusmenetelmän toteuttamiseen ei ole sinällään erityisen uusi idea. Vastaavanlaista on kokeiltu esimerkiksi siipiprofiilien adaptiivisessa laskennassa, jossa adaptiivisuutta on ajettu turbulenssisuureisiin pohjautuvalla virheindikaattorilla, ja vaikka ratkaisija on alun perin kehitetty enimmäkseen liikkuville eli dynaamisille laskentaverkoille, myös adaptiivisuus on mahdollista toteuttaa. (Lindblad 2014, s. 3–60). Vastaavanlainen *pimpleDyMFoam*-pohjainen ja 2D-simulaatioihin sopiva alkeellinen adaptiivinen ratkaisija löytyy esimerkiksi lähteestä (Holzmann 2017). Ongelmana virallisessa ja OpenFOAM-yhteisön tuottamissa ratkaisijoissa on se, että näissä kaikissa adaptiivisuutta ajavat mielivaltaiset virtauskentän suureet, eikä näitä siten voida pitää todellisina adaptiivisina ratkaisijoina. Adaptiivisuuden topologia-aspekti on jo selkeästi saatu kehitettyä hyvälle tasolle ja muutamaan otteeseen todettu toimivaksi, mutta virhearvioita tuottavat lisäominaisuudet on kuitenkin toteutettava alusta lähtien tai tuotava ne OpenFOAM:iin jostakin muusta sopivasta lähteestä.

Virhearvioiden toteuttamista helpottaa merkittävästi OpenFOAM:iin ajoittain julkaistava *foam-extend* -työkalupakki. Esimerkiksi *foam-extend*in version 4.0 käyttäjän käsikirjaan (2009, s. 93–94) on listattu muutamia käyttökelpoisia laminaaristen virtausten jälkikäsitelyyn sopivia virhearviotyökaluja, joilla voidaan luoda momenttimenetelmään ja jäänösmenetelmään pohjautuvia virhearvioita. Nämä eivät sellaisenaan sovi turbulentsille virtaustapauksille, koska virhearvioissa on otettava huomioon myös turbulenssin aiheuttamat lisätermiit. Toisaalta pelkkä tulosten jälkikäsitely ei riitä, vaan työkalut on muutettava sellaiseen muotoon, että näiden käyttäminen onnistuu myös simulaation aikana.

Tavoitteena on siis toteuttaa kuvan 4.1 kaltainen vuokaaviona esitetty sovellus, joka hyödyntää PIMPLE-algoritmia ja laskennan päätteeksi suorittaa virhearvion sekä tämän perusteella määrittää, missä laskentaverkkoa kannattaa tihentää tai harventaa. Ajatuksena on ratkaista SIMPLE-algoritilla aina stationääritila ja sen jälkeen marssittaa aikaa aina yhden aika-askeleen verran eteenpäin. Varsinaisesti aikahistoriaa ei ole tarkoitus ratkaista, vaan aika-askeleena käytetään pseudoaikaa. Myöskään PISO-algoritmin mukaisia useampia nopeuskorjauksia ei suoriteta, vaan kuvassa 4.1 näkyvä paineyhtälön ratkaiseminen ja nopeuskentän korjaaminen suoritetaan vain kertaalleen.



Kuva 4.1 Adaptiivinen PIMPLE-algoritmi

PimpleDyMFoam-ratkaisijassa violetilla reunustetut vaiheet ovat täysin käyttäjän vastuulla. Käyttäjän generoimaan laskentaverkkoon ja laskettavien suureiden alkuarvaukseen ei ole tarkoitus tehdä sen kummempaa automatisointia, koska nämä ovat aina tilanteesta riippuvia. Keskitytään siis seuraavaksi sinisellä laatikoitujen osuuksien toteuttamiseen ja myöhemmin luvussa 4.2 itse virhearviokohdan kehittämiseen.

OpenFOAM:issa dynaaminen laskentaverkko voidaan luoda usealla eri tavalla. *DynamicFvMesh*-kirjastoa voidaan hyödyntää ratkaisijoissa muun muassa mahdollistamaan jäykän kappaleen liike *solidBodyMotionFvMesh*-luokalla. Laskentaverkon muotoa voidaan myös manipuloida tai lisätä siihen liikettä esimerkiksi *dynamicInkJetFvMesh*- ja *motionSolverFvMesh*-luokilla. Laskentaverkon koon muuttamiseen on kuitenkin tehty erillinen *dynamicRefineFvMesh*-luokka, joka kykenee tihentämään ja harventamaan laskentaverkkoa käyttäjän asettamien raja-arvojen perusteella. Tämän lisäksi kyseinen luokka mahdollistaa edellisen laskentaverkon tuloksien interpoloinnin uudelle laskentaverkolle. Luokka sisältää myös muita hyödyllisiä ominaisuuksia, kuten montako kertaa yksittäistä kontrollitilavuutta voidaan pilkkoa pienemmäksi ja halutaanko liiallisia hypäyksiä laskentaverkon koossa kontrollitilavuuksien välillä rajoittaa. Luokan ominaisuudet voidaan ottaa käyttöön erillisellä *dynamicMeshDict*-sanakirjalla. (Eltard-Larsen 2016, s. 1–10) Edellä mainitun *dynamicRefineFvMesh*-luokan käyttöä on havainnollistettu ohjelmassa 4.1, jonka avulla on mahdollista toteuttaa tarvittu adaptiivisuuteen liittyvät ominaisuudet. Käyttäjälle jää tehtäväksi vain asettaa virhearvion perusteella *lowerRefineLevel*- ja *upperRefineLevel*-kynnysarvot, joiden sisällä kontrollitilavuudet pilkotaan pienemmiksi ja vastaavasti jokin *unrefineLevel*-alaraja, jonka perusteella laskentaverkkoa harvennetaan kriteerin täyttävissä kohdissa.

```
dynamicFvMesh    dynamicRefineFvMesh; // Dynaamisen luokan valinta

dynamicRefineFvMeshCoeffs
{
    refineInterval    1;           // Miten usein adaptiivisuutta käytetään
    field             momErrorU;  // Adaptiivisuudessa käytettävä suure
    lowerRefineLevel  1.295;      // Alempi hienontamisen raja-arvo
    upperRefineLevel  99999;      // Ylempi hienontamisen raja-arvo
    unrefineLevel     0.432;      // Karkeammaksi muuttamisen raja-arvo
    nBufferLayers     5;          // Rajapintasolujen asettaminen
    maxRefinement     2;          // Maks. hienontamistasojen lukumäärä
    maxCells          15000000;   // Laskentaverkon maksimikoko
    correctFluxes     // Interpoloitavat suureet
    (
        (phi U)           // Nopeuskentän interpolointi
    );
    dumpLevel         true;       // Tallentaa laskentaverkon hienontamistason
}
```

Ohjelma 4.1 *dynamicRefineFvMesh*-luokan säätöparametrit

Tarkempi katsaus OpenFOAM:in *dynamicRefineFvMesh*-luokan lähdekoodiin (2019) paljastaa, että laskentaverkon muokkaaminen voidaan suorittaa vain käyttämällä skalarisuureita, eli toisin sanoen vektorisuureiden käyttö ei ole mahdollista. Sisäänrakennettu

ominaisuus sallii vain heksaedrin muotoisten kontrollitulavuuksien leikkaamiseen kahdeksaan pienempään osaan. Laskentaverkon anisotrooppinen leikkaaminen on periaatteessa mahdollista toteuttaa OpenFOAM:iin, mutta valmiita testattuja kirjastoja ei ole saatavilla, joten työssä hyödynnetään vain isotrooppista leikkaamista. Tarkempaa tietoa toimivan anisotrooppisen kirjaston toteuttamisesta on lähteessä (Karlsson 2012).

Toteutettavaan ohjelmaan jää siis jäljelle vain virhearviokohdan täyttäminen. PIMPLE-algoritmi osaa hoitaa kuvan 4.1 mustalla reunalla olevat perustoimenpiteet, ja *dynamicRefineFvMesh*-luokka suorittaa sinisellä laatikoidut vaiheet. Keskitytään seuraavaksi siihen, miten virhearviot voidaan lisätä valmiina olevaan *pimpleDyMFoam*-ratkaisijaan.

4.2 Lisäominaisuuksien ohjelmointi

Työssä ratkaistavien RANS-yhtälöiden (2.21) virhe-estimaatit voidaan toteuttaa ohjelmalla virhearvioiden sisältämät termit yksi kerrallaan, mutta konvektiotermin tapauksessa täytyy kuitenkin ottaa huomioon eräs virtauslaskennassa käytetty stabilointikeino. OpenFOAM:in käyttäjän käsikirjan (Greenshields 2017, s. 122) mukaan virtauskentän stationääritilaa ratkaistaessa yleensä diskretisoidaan materiaalsen aikaderivaatan molemmat osuudet, eli yleisemmin

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = \frac{\partial\phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) - (\nabla \cdot \mathbf{u})\phi, \quad (4.1)$$

koska tämän avulla ratkaisu saadaan suppenemaan nopeammin ja termin vaikutus ideaalitalapauksissa häviää kokonaan, jos nopeuskenttä on täysin divergenssitön. Lisätermin ansiosta RANS-yhtälöiden (2.21) todellinen ratkaistava vektorimuotoinen yhtälö on

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - (\nabla \cdot \mathbf{u}) \cdot \mathbf{u} = -\nabla p + \nabla \cdot (\nu\nabla\mathbf{u}) + \nabla \cdot \boldsymbol{\tau}, \quad (4.2)$$

jonka sisältämän uuden termin vaikutus on otettava virhearvioissa huomioon. Itseasiassa uusi termi on implisiittinen lähde-termi, koska tämän suuruus riippuu myös vektorista \mathbf{u} . Loput lähde-termeistä, eli painegradientti ja Reynoldsin jännitystensorin divergenssi ovat molemmat eksplisiittisiä. Momenttijäännös (3.36) voidaan laajentaa vektorimuotoiseksi ja lopputuloksena saadaan RANS-yhtälöiden momenttijäännökseksi

$$\begin{aligned} \bar{r}_{m,c}(m^{\mathbf{u}}) = \int_{V_c} [\nabla \cdot (\mathbf{u}m^{\mathbf{u}}) - \nabla \cdot (\nu\nabla m^{\mathbf{u}}) - (-\nabla p) \cdot \mathbf{u} - (\nabla \cdot \boldsymbol{\tau}) \cdot \mathbf{u} - 2(\nabla \cdot \mathbf{u})m^{\mathbf{u}} \\ + \nu(\nabla\mathbf{u} : \nabla\mathbf{u})]dV, \end{aligned} \quad (4.3)$$

sekä vastaavalla tavalla saadaan muodostettua RANS-yhtälöiden jäännösvirheeksi

$$\bar{r}_{r,c}(\mathbf{u}) = \int_{V_c} [\nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot (\nu\nabla\mathbf{u}) - (-\nabla p) - (\nabla \cdot \boldsymbol{\tau}) - (\nabla \cdot \mathbf{u}) \cdot \mathbf{u}]dV, \quad (4.4)$$

jotka voidaan normalisoida lukujen 3.3 ja 3.4 mukaisesti ja tämän jälkeen ohjelmoida *pimpleDyMFoam*-ratkaisimeen. Momenttivirhe-estimaatin formuloinnissa implisiittisen lähdetermin kohdalla on hyödynnetty riippuvuutta $m^u = 1/2\|\mathbf{u}\|^2$, joka aiheuttaa kertoimen 2,0 kyseisen termin eteen.

OpenFOAM:in ohjelmoijan käsikirjan (2018, s. 35–39) mukaan Laplacen termien, konvektiitermien, gradienttien ja implisiittisten lähdetermien ohjelmointi onnistuu suoraan käyttämällä sisäänrakennettuja funktioita *fv::laplacian()*, *fv::div()*, *fv::grad()* ja *fv::Sp()*, ja eksplisiittiset lähdetermit on tapana toteuttaa sellaisenaan tai käyttää automaattista funktiota *fv::SuSp()*, joka ohjelmoi lähdetermin implisiittiseksi tai eksplisiittiseksi sen merkin perusteella. Hyödyntämällä kyseisiä funktioita, momenttivirhe-estimaatti voidaan muodostaa ohjelman 4.2 kaltaisella koodilla, jonka perustana käytettiin laminaarista *icoMomentError*-työkalua foam-extend 4.0 -työkalupakista.

```

volScalarField momError // Momenttivirheobjekti
(
    IObject
    (
        "momError" + U.name(),
        mesh.time().timeName(),
        mesh,
        IObject::NO_READ,
        IObject::NO_WRITE
    ),
    sqrt
    (
        2.0*mag
        (
            (
                fvc::div(phi, mU) // Divergenssitermi
                - nuEff*fvc::laplacian(mU) // Laplacen termi
                - (U & -fvc::grad(p)) // 1. Ekspl. lähdetermi
                - (U & fvc::div(turbulence->R())) // 2. Ekspl. lähdetermi
                - 2.0*fvc::Sp(fvc::div(U), mU) // Impl. lähdetermi
                + 0.5*nuEff*
                (
                    gradU && (gradU + gradU.T())
                )
            )
            )*L/(mag(U) + nuEff/L) // Normalisointikerroin
        )
    );
momError.boundaryFieldRef() = 0.0; // Asettaa reunoilla virheen nollaksi
momError.write(); // Tallentaa momenttivirheen

// Momenttivirheen maksimin, keskiarvon ja tilavuuskeskiarvon tulostaminen
Info<< "\n momError max: "
<< max(momError).value() << endl;
Info<< "\n momError mean: "
<< momError.average().value() << endl;
Info<< "\n momError volMean: "
<< momError.weightedAverage(mesh.V()).value() << endl;

```

Ohjelma 4.2 Momenttivirhe-estimaatin OpenFOAM-toteutus

Viimeisen termin muoto ohjelmassa 4.2 on yhtälöstä (4.3) hieman poikkeava, mutta kyseistä muotoa päädyttiin käyttämään, koska näin oli tehty myös momenttivirhe-estimaatin pohjana olevassa työkalussa alun perin. Eroavaisuus havaittiin joka tapauksessa varsin pieneksi. Momenttivirhe-estimaatissa tarvittavat suureet, kuten efektiivinen viskositeetti, eli kinemaattisen viskositeetin ja pyörreviskositeetin summa, saadaan muodostettua ennen momenttivirhe-estimaattia esimerkiksi ohjelman 4.3 mukaisella tavalla. Vastaavasti nopeusvektorin toinen momentti, nopeusgradientti ja normalisoinnissa tarvittava laskentaverkon karakteristinen mitta kannattaa määrittää samassa ohjelmassa.

```

volScalarField nuEff(turbulence->nut() + turbulence->nu()); // Efektiivinen
                                                         viskositeetti
volScalarField mU = 0.5*magSqr(U); // Nopeusvektorin toinen momentti
volTensorField gradU = fvc::grad(U); // Nopeusgradientti

volScalarField L // Karakteristisen mitan objekti
(
    IObject
    (
        "L",
        mesh.time().timeName(),
        mesh,
        IObject::NO_READ,
        IObject::NO_WRITE
    ),
    mesh,
    dimensionedScalar("one", dimLength, 1.0)
);
L.ref() = // Karakteristisen mitan laskenta
          mesh.V()/fvc::surfaceSum(mesh.magSf())().internalField();

```

Ohjelma 4.3 Momenttivirhe-estimaatin muut suureet

Jäännösvirhe-estimaatti on huomattavasti hankalampi toteuttaa, koska OpenFOAM:issa ei ole suoraan saatavilla tähän tarvittavia ominaisuuksia. Tästä syystä foam-extend 4.0 -työkalupakista joudutaan hakemaan kokonainen *errorEstimation*-kirjasto ja soveltamaan sen jälkeen tästä löytyviä ominaisuuksia. Valitettavasti foam-extend 4.0 -työkalupakin kirjastot eivät suoraan toimi OpenFOAM:in uusimmissa versioissa, sillä OpenFOAM v4.0 -muutoslokin (2016) mukaan tästä versiosta lähtien sovelluksen lähdekoodissa osoittimien toimintaa on muutettu, joten haettuun kirjastoon on tehtävä varsin paljon muutoksia osoittimien osalta, jotta yhteensopivuus saadaan palautettua. Tarvittavien muutosten lista on pitkä ja prosessi varsin työläs, joten sitä ei tässä erikseen esitetä. Kirjaston koodi on sinällään toimivaa lähes sellaisenaan, vaan ongelma liittyy funktioiden *internalField()* ja *boundaryField()* osoittimiin, jotka pitää päivittää uuden version mukaisiksi.

Kirjaston muokkaamisen jälkeen voidaan jäännösvirhe-estimaatti kätevästi ohjelmoida käyttämällä kirjaston funktioita *resError::laplacian()*, *resError::div()* ja *resError::Sp()* ohjelman 4.4 esittämällä tavalla. Eksplisiittiset lähdeterminit voidaan toteuttaa OpenFOAM:in omilla sisäänrakennetuilla käskyillä samalla tavalla kuin momenttivirhemenetelmän yhteydessä tehtiin.

```

errorEstimate<vector> rr      // Jäännösvirhevektori
(
    resError::div(phi, U)      // Divergenssitermi
    -resError::laplacian(nuEff, U) // Laplacen termi
    -resError::Sp(fvc::div(phi), U) // Impl. lähdetermi
    ==
    -fvc::grad(p)             // 1. Ekspl. lähdetermi
    +fvc::div(turbulence->R()) // 2. Ekspl. lähdetermi
);

volVectorField resErrorU = rr.error(); // Palauttaa normalisoidun
                                        jäännösvektorin
volScalarField magResErrorU = mag(resErrorU); // Laskee jäännösvektorin
                                                suuruuden
resErrorU.write(); // Tallentaa jäännösvektorin
magResErrorU.write(); // Tallentaa jäännösvektorin
                        suuruuden

// Jäännösvirheen maksimin, keskiarvon ja tilavuuskeskiarvon tulostaminen
Info<< "\n    resError max: "
<< max(magResErrorU).value() << endl;
Info<< "\n    resError mean: "
<< magResErrorU.average().value() << endl;
Info<< "\n    resError volMean: "
<< magResErrorU.weightedAverage(mesh.V()).value() << endl;

```

Ohjelma 4.4 Jäännösvirhe-estimaatin OpenFOAM-toteutus

Tarkempi katsaus foam-extendin *errorEstimation*-kirjaston lähdekoodin (2019) sisältämiin funktioihin osoittaa, että divergenssitermin diskretisointivirheen arvioiminen tapahtuu funktion *resError::div()* sisällä interpoloimalla nopeusvektorin arvot lineaarisen variaation avulla jokaiselle rajapinnalle. Tämän jälkeen jokaiselle pinnalle muodostetaan konvektiovuot ja summaamalla nämä saadaan kontrollitilavuuskohtainen divergenssijäännös. Vastaavasti Laplacen termin diskretisointivirhe arvioidaan funktion *resError::laplacian()* sisällä rajapinnoille muodostettujen gradienttien avulla, kun näillä muodostetaan diffuusiovuot rajapinnoille ja summaamalla nämä vuot yhteen saadaan selville Laplacen termin jäännös. Samalla molemmat funktiot määrittävät tarvittavat kontrollitilavuuskohtaiset normalisointikertoimet. Implisiittisen lähdetermin tapauksessa funktio *resError::Sp()* lisää lähdetermin sellaisenaan jäännökseen ja samalla myös normalisointikertoimeen. Eksplisiittiset lähdetermit on toteutettu ohjelmaan 4.4 sellaisenaan, eikä niillä siten ole vaikutusta normalisointikertoimeen, kuten luvussa 3.4 tuotiin esille.

Lopuksi ohjelmassa 4.4 esiintyvä funktio *error()* palauttaa jäännösvirhe-estimaatin jakamalla jokaista kontrollitilavuutta kohden tunnetut jäännökset näille muodostetuilla normalisointikertoimilla. Ohjelmassa lasketaan vielä erikseen jäännösvirhevektorin L1-normi, sillä *pimpleDyMFoam*-ratkaisija osaa muuttaa laskentaverkon kokoa vain skalaarisuureiden pohjalta. Jäännösvirheen normi voidaan laskea muullakin tavalla, mutta L1-normia käytetään myös foam-extend 4.0 -ohjelman *icoErrorEstimate*-työkalussa, jota käytettiin ohjelman 4.4 pohjana, eikä sen vuoksi normia lähdetty enää vaihtamaan.

4.3 Laskenta-asetusten valinta

Täten tarvittavat ominaisuudet on saatu toteutettua ja näillä voidaan määrittää laskettujen tulosten diskretisointivirheet. Käydään läpi vielä seuraavaksi, miten teorialuvussa esillä olleet diskretisointimenetelmät ja ratkaisualgoritmit voidaan kytkeä laskennassa päälle.

OpenFOAM:issa diskretisointimenetelmien valitseminen tapahtuu luomalla *fvSchemes*-sanakirja, jolla aktivoidaan yhtälöiden ratkaisemisessa tarvittavat diskretisointimenetelmät. Diskretisointimenetelmät ovat täysin vapaasti valittavissa, mutta muutamia suosituksia on olemassa. Esimerkiksi suureiden interpoloinnissa kontrollitulavuuksien rajapinnoille kannattaa käyttää lineaarista interpolointia, vaikka periaatteessa muitakin vaihtoehtoja on. Edellä mainitun lisäksi stationääritilaan pyrkivissä simulaatioissa divergenssi-termien ylimääräinen osa kannattaa ottaa huomioon valitsemalla divergenssimenetelmistä *bounded*-variantti, jotka käsittelevät tämän lisätermin automaattisesti. Laskentaverkon topologian asianmukainen huomioonottaminen on käyttäjän vastuulla, mutta yleisesti epäortogonaalisen gradienttikorjauksen suuruutta kannattaa rajoittaa, jos laskentaverkon maksimaalinen epäortogonaalisuus ylittää 70° . (Greenshields 2017, s. 118–125)

Laskentaa varten aikaintegrointimenetelmäksi valittiin *steadyState*, eli tällöin aikaderivaatat asetetaan nolllaksi, eikä näitä erikseen ratkaista. Painotettu pienimmän neliösumman gradienttimenetelmä kytkettiin päälle kaikkiin gradientteihin valitsemalla *leastSquares*-vaihtoehto. Divergenssitermeihin jouduttiin kiinnittämään enemmän huomiota, sillä ratkaistavissa yhtälöissä esiintyy sekä vektorimuotoisia että skalaarimuotoisia divergenssitermejä. Nopeuden divergenssille valittiin menetelmäksi *bounded Gauss GammaV 1.0* ja muille suureille *bounded Gauss Gamma 1.0*. Reynoldsin jännitystensorin divergenssin diskretisointi vaatii keskeisdifferenssimenetelmän, sillä kyseiselle termille on saatavilla vain *Gauss linear* -menetelmä. Lisäksi nopeusvektorin toiselle momentille ja nopeusvektorin divergenssille asetettiin virhearvioita varten *Gauss linear* -menetelmä. OpenFOAM:in lähdekoodi (2019) osoittaa, että Gamma-menetelmien skalaariparametri 1,0 todellisuudessa vastaa teorian mukaista arvoa 0,5, jonka Gamma-menetelmän yhteydessä todettiin olevan paras vaihtoehto stationääritilan ratkaisuja haettaessa.

Laplacen termeille valittiin menetelmäksi *Gauss linear limited 0.5*, joka kytkee päälle alirelaksoidun epäortogonaalisen lisäkorjauksen kaikille Laplacen termeille ja samalla rajoittaa epäortogonaalisen korjauksen suuruutta. Poikkeuksellisesti momenttinvirhe-estimaattia varten nopeusvektorin toisen momentin Laplacen termille asetettiin *Gauss linear corrected*, joka sallii täyden epäortogonaalisen lisäkorjauksen. Suureiden interpolointiin kontrollitulavuuden sisällä valittiin suositusten mukaisesti *linear*. Pinna normaalin suuntaisen gradientin laskemista varten asetettiin *limited 0.5*, jotta diskretisointi on konsistentti myös Laplacen termeihin nähden. Itseasiassa Greenshieldsin (2017, s. 123–125) mukaan arvo 0,5 rajoittaa epäortogonaalista korjausta siten, että tämän suuruus voi olla korkeintaan lasketun ortogonaalisen gradientin suuruinen. Valitut diskretisointimenetelmät on esitetty kokonaisuudessaan ohjelmassa 4.5

```

ddtSchemes          // Aikaintegrointimenetelmät
{
    default          steadyState;
}
gradSchemes         // Gradienttimenetelmät
{
    default          leastSquares;
}
divSchemes          // Divergenssimenetelmät
{
    default          none;
    div(phi,U)      bounded Gauss GammaV 1.0;
    div(phi,k)      bounded Gauss Gamma 1.0;
    div(phi,omega)  bounded Gauss Gamma 1.0;
    div((nuEff*dev2(T(grad(U)))) Gauss linear;
    div(R)          Gauss linear; // Virhearvioille
    div(U)          Gauss linear; // Virhearvioille
    div(phi,(0.5*magSqr(U))) Gauss linear; // Virhearvioille
}
laplacianSchemes   // Laplacen menetelmät
{
    default          Gauss linear limited 0.5;
    laplacian((0.5*magSqr(U))) Gauss linear corrected; // Virhearvioille
}
interpolationSchemes // Interpolointimenetelmät
{
    default          linear;
}
snGradSchemes      // Pinnan normaalin suuntaiset gradienttimenetelmät
{
    default          limited 0.5;
}
fluxRequired       // Vuoaasetukset
{
    default          no;
    pcorr;
    p;
}
wallDist           // Seinämätäisyyden laskentamenetelmät
{
    method meshWave;
}

```

Ohjelma 4.5 Diskretisointimenetelmien valinta *fvSchemes*-sanakirjassa

Ratkaisualgoritmiin, alirelaksointiin ja matriisilaskennan menetelmiin liittyvät ominaisuudet voidaan kytkeä päälle OpenFOAM:issa *fvSolution*-sanakirjalla ja erityisesti PIMPLE-algoritmin hienosäätämiseen on kehitetty erilaisia valintoja. Esimerkiksi algoritmin asetuksissa voidaan säätää *nCorrectors*-parametria, jolla voidaan määrittää PISO-iteraatiokierrosten lukumäärä ja vastaavasti ulkoiteraatioiden, eli SIMPLE-iteraatioiden lukumäärä määritetään *nOuterCorrectors*-parametrin avulla. PIMPLE-algoritmiin on myös liiallisen epäortogonaalisuuden vuoksi mahdollista tehdä vielä ylimääräisiä paineenkorjausiteraatioita *nNonOrthogonalCorrectors*-käskyllä, mutta tyypillisesti näitä ei kuitenkaan suositella tehtäväksi. Lisäksi *momentumPredictor*-säätö mahdollistaa vielä ylimääräisen lisäyksen algoritmiin. Alirelaksoinnin osalta on mahdollista käyttää joko

eksplisiittistä relaxointia kenttämuuttujille tai Patankarin matriisimuotoista alirelaksointia ratkaistaville yhtälöille. (Greenshields 2017, s. 125–131)

PIMPLE-algoritmin sisällä valittiin konsistentti SIMPLEC-versio SIMPLE-algortmistista ja ulkoiteraatioiden maksimimääräksi asetettiin 500 kappaletta. SIMPLEC-algoritmin sisällä paineenkorjaus suoritetaan vain kertaalleen, eli tämän jälkeen erillisiä PISO-kierroksia ei enää tehdä. Erillisiä laskentaa stabiloivia epäortogonaalisia lisäkierroksia ei lisätty laskentaan lainkaan, koska näillä ei ollut merkittävää vaikutusta laskentaan. PIMPLE-algoritmi asetettiin pysähtyvän, kun jäännökset tippuvat alle 10^{-3} toleranssin tai iteraatioiden maksimimäärä täyttyy. Valitut PIMPLE-algoritmin säädöt on esitetty ohjelmassa 4.5.

```
PIMPLE
{
    correctPhi                yes; // Dynaamisille laskentaverkoille
    consistent                yes; // SIMPLEC-variantin valinta
    turbOnFinalIterOnly      no;  // Turb. yhtälöiden ratkaiseminen
    nOuterCorrectors         500; // Ulkoiteraatioiden määrä
    nCorrectors               1;   // Paineenkorjausiteraatioiden määrä
    nNonOrthogonalCorrectors 0;   // Ep. ortog. lisäiteraatiot
    residualControl          // Toleranssien käyttö
    {
        "(U|p|k|omega)"      // Sovellettavat suureet
        {
            relTol            0;   // Suhteellinen toleranssi
            tolerance         1e-3; // Absoluuttinen toleranssi
        }
    }
}
```

Ohjelma 4.5 PIMPLE-algoritmin säätöparametrit *fvSolution*-sanakirjassa

Alirelaksointikertoimien kohdalla meneteltiin sillä tavalla, että vain painekentälle määritettiin eksplisiittinen alirelaksointikerroin. Kertoimeksi päädyttiin valitsemaan 0,9, koska tämä oli suurin mahdollinen kerroin, joka mahdollisti laskennan olevan stabiili seuraavassa luvussa tutkittavalle virtausmallille. Teoriassa SIMPLEC-variantti ei vaadi painekentälle lainkaan alirelaksointia, mutta todellisuudessa pieni alirelaksointi on paikallaan ja tämä stabiloii laskentaa merkittävästi. Nopeuskentän yhtälöille asetettiin alirelaksointikertoimeksi 0,9, joka mahdollisti mahdollisimman nopean suppenemisen. Sen sijaan turbulenssisuureille jouduttiin laskemaan alirelaksointikertoimia arvoon 0,7, jotta laskenta säilyi stabiilina. Relaxointikertoimien asettamista on havainnollistettu ohjelmassa 4.6.

Lineaarialgebran yhtälöiden ratkaisemiseen OpenFOAM tarjoaa suuren määrän erilaisia matriisilaskentaa kehitettyjä ratkaisijoita, esimerkiksi multigrid-menetelmiä erilaisilla tasoittimilla ja useampia pohjustettuja konjugaattigradienttimenetelmiä. Lineaarialgebran ratkaisijoiden toimintaan voidaan vaikuttaa säätämällä niiden parametreja, toleranssia, suhteellista toleranssia, eli alkujäännöksen ja tavoitellun loppujäännöksen suhdetta, sekä iteraatioiden maksimimäärää. (Greenshields 2017, s. 125–129)

```

relaxationFactors      // Alirelaksointikertoimet
{
    fields              // Eksplisiittinen alirelaksointi kenttäsuureille
    {
        "(p|pFinal)"    0.9;
    }
    equations           // Matriisimuotoinen alirelaksointi yhtälöille
    {
        "(U|UFinal)"    0.9;
        "(k|kFinal)"    0.7;
        "(omega|omegaFinal)" 0.7;
    }
}

```

Ohjelma 4.6 Alirelaksointiin liittyvät parametrit *fvSolution*-sanakirjassa

Paine yhtälöiden ratkaisemisessa hyödynnettiin *GAMG*-ratkaisijaa, jonka suorituskyvyn havaittiin olevan paras kaikista menetelmistä, kun tasoittimeksi valittiin *DICGaussSeidel*. Liiketyhtälöiden ja turbulenssiyhtälöiden ratkaisemisessa käytettiin *PBiCGStab*-ratkaisijaa *DILU*-pohjustimella, joka myös osoittautui parhaaksi vaihtoehdoksi. Kaikkien pysähtymistoleransseiksi annettiin 10^{-6} eikä iteraatioiden maksimimäärää rajoitettu tai suhteellista toleranssia käytetty. Valitut lineaarialgebran ratkaisijat on esitetty ohjelmassa 4.6.

```

"(pcorr|pcorrFinal|p|pFinal)"
{
    solver              GAMG;           // Valittu ratkaisija
    tolerance           1e-6;          // Absoluuttinen toleranssi
    relTol              0;             // Suhteellinen toleranssi
    smoother            DICGaussSeidel; // Käytettävä tasoitin
    nPreSweeps          0;             // Ratkaisijan säätöparametri
    nPostSweeps         2;             // Ratkaisijan säätöparametri
    cacheAgglomeration true;          // Ratkaisijan säätöparametri
    agglomerator        faceAreaPair;  // Ratkaisijan säätöparametri
    nCellsInCoarsestLevel 10;         // Ratkaisijan säätöparametri
    mergeLevels         1;            // Ratkaisijan säätöparametri
}

"(U|UFinal|k|kFinal|omega|omegaFinal)"
{
    solver              PBiCGStab;     // Valittu ratkaisija
    preconditioner      DILU;          // Käytettävä pohjustin
    tolerance           1e-6;          // Absoluuttinen toleranssi
    relTol              0;             // Suhteellinen toleranssi
}

```

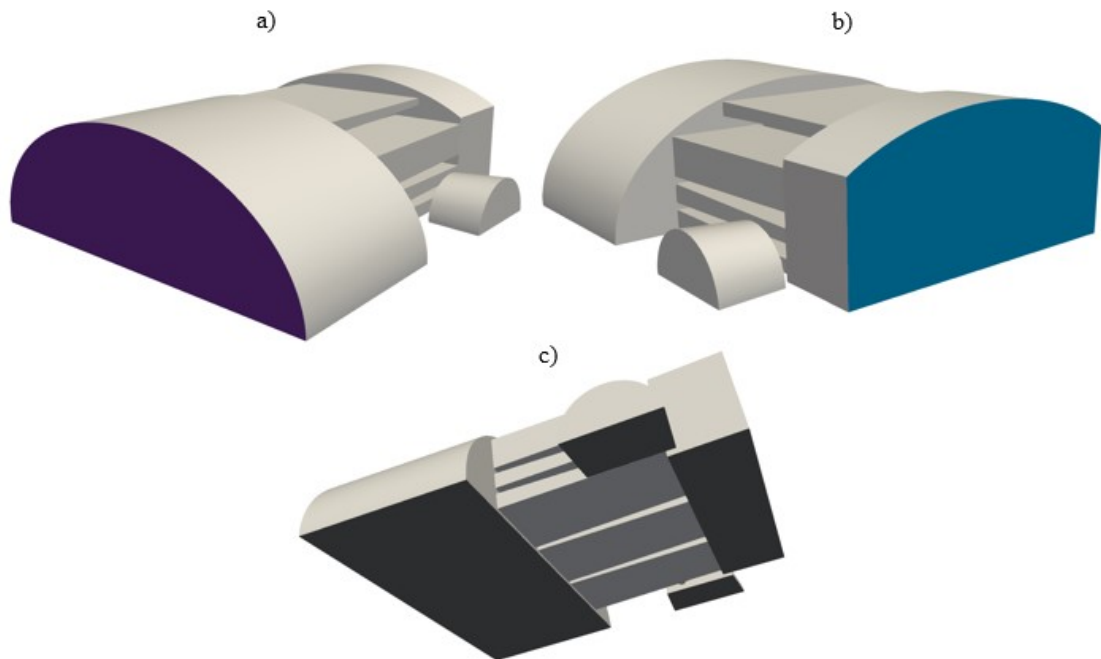
Ohjelma 4.6 Lineaarialgebran ratkaisijat *fvSolution*-sanakirjassa

Nyt kaikki tarvittavat säädöt on tehty ja tarvittavat ominaisuudet toteutettu *pimpleDyM-Foam*-ratkaisijaan, joten seuraavaksi voidaan keskittyä tutkimaan h-adaptiivisen algoritmin toimintaa käytännön insinöörisovelluksessa ja tämän jälkeen voidaan arvioida laskentatulosten virhetasoa valituilla virhearviomenetelmillä. Kaikki edellä esitetyt asetukset on pidetty samanlaisena koko työn läpi, jotta tulokset ovat vertailukelpoisia keskenään, ja simulaatiot ajettiin käyttäen neljää rinnakkaista prosessia, joita varten laskenta-alue jaettiin *OpenFOAM*:in sisäänrakennetulla *scotch*-työkalulla neljään osaan.

5. TOTEUTETUN MENETELMÄN KÄYTÄNNÖN-SOVELLUS

5.1 Esimerkkitapaus ja ongelman asettelu

Kehitetyn adaptiivisen *pimpleDyMFoam*-ratkaisijan toimintaa päädyttiin testaamaan kuvan 5.1 mukaisessa insinöörisovelluksessa, jonka tavoitteena on mallintaa staattisen pitotputken periaatteella toimivan virtausmittarin sisälle jäävää ilmatilavuutta ja ratkaista siihen muodostuva virtauskenttä mahdollisimman tarkasti. Virtausmittarissa ilmavirtaus pakotetaan elliptisen muotoisesta sisääntulosta rakovirtaukseksi, jonka jälkeen ilmavirta ohjataan takaisin suurempaan tilavuuteen. Geometria on symmetrinen korkeussuunnassa, joten geometria on leikattu tällä symmetriatasolla ennen laskentaverkon luomista.



Kuva 5.1 Testigeometrian a) sisäänmeno, b) ulostulo ja c) symmetriataso

Laskentaa varten muodostettiin mahdollisimman harva laskentaverkko, joka kykenee kuvaamaan virtausmittarin sisägeometriaa riittävällä tarkkuudella. Laskentaverkko päädyttiin generoimaan foam-extend 4.0 -työkalupakista löytyvällä *cfMesh*-verkotustyökalulla, jolla muodostettiin pääasiassa tasakokoisista heksaedrien muotoisista kontrollitilavuuksista oleva laskentaverkko. Kuvassa 5.1 näkyvien sylinterimäisten mittakammioiden ja rakojen välillä olevien ohuiden kanavien ympäristö tihennettiin vielä erikseen. Kontrollitilavuuksia on alkuarvauksena käytettävässä laskentaverkossa yhteensä noin 737 000 kappaletta. Tämän jälkeen simulaatiossa laskettaville suureille asetettiin taulukon 5.1 mukaiset reunaehdot. Asetetuista reunaehdoista on muutama sana paikallaan.

OpenFOAM:issa luonnollisella *zeroGradient*-reunaehdolla voidaan asettaa käyttäjän valitsemille pinnoille haluttujen suureiden pinnan normaalin suuntainen gradientti nolllaksi ja *fixedValue*-reunaehdot puolestaan käytetään asettamaan kyseiselle pinnoille jokin käyttäjän valitsema oleellinen reunaehto. Näiden yhdistelmänä olevaa *inletOutlet*-reunaehdot hyödynnetään yleensä tapauksissa, joissa reuna halutaan asettaa *zeroGradient*-tyyppiseksi, kun kyseisellä reunalla virtaus on pinnan normaalin suuntaista, ja *fixedValue*-tyyppiseksi, jos virtaus kulkee vastakkaiseen suuntaan, eli ilmenee takaisinvirtausta. Kokonaispaineen $p = p_0 - 1/2\|\mathbf{u}\|^2$ asettavaa *totalPressure*-reunaehdot käytetään yleisesti *pressureInletOutletVelocity*-reunaehdon kanssa, joiden ideana on estää liian suuria nopeusvaihteluita laskennan edetessä, koska virtausta hallitsee pääasiassa sen painegradientti, eikä nopeuskenttä täten pääse hallitsemattomasti kasvamaan. *Symmetry*-reunaehdot taas puolestaan peilaa geometrian asetetulla reunalla pinnan normaalin suuntaisesti. (Greenshields 2017, s. 141–144) Kokonaispaineen arvoksi asetettiin sisäänmenon ja ulostulon välillä, joka aiheuttaa virtauksen ulostulon suuntaan. Ulostulossa nopeuden *inletOutlet*-reunaehdon avulla esitettiin takaisinvirtaus kokonaan ja turbulenssisuureille käytettiin takaisinvirtaukselle mitättömän pieniä arvoja, koska esimerkiksi ω -suureen asettaminen nolllaksi saattaa johtaa nolllalla jakamiseen, sillä pyörreviskositeetti määräytyy pääasiassa suhteesta $\nu_t = k/\omega$.

Taulukko 5.1 Laskennassa käytetyt reunaehdot

Suure	Sisäänmeno	Ulostulo	Seinämät	Symmetriataso
p (m ² /s ²)	totalPressure	fixedValue	zeroGradient	symmetry
\mathbf{u} (m/s)	pressureInletOutletVelocity	inletOutlet	noSlip	symmetry
k (J/kg)	turbulentIntensityKineticEnergyInlet	inletOutlet	kLowReWallFunction	symmetry
ω (1/s)	turbulentMixingLengthFrequencyInlet	inletOutlet	omegaWallFunction	symmetry
ν_t (m ² /s)	calculated	calculated	nutUSpaldingWallFunction	symmetry

Turbulenssisuureiden ja pyörreviskositeetin seinäfunktioita (engl. wall function) käytetään OpenFOAM:issa yleensä sen vuoksi, että ilman näitä jokaisella seinällä seinänviereisen kontrollitilavuuden keskipisteen tulisi sijaita viskoosin rajakerroksen sisällä, jotta turbulenssimallit toimisivat hyvin. Tästä syystä joudutaan usein käyttämään kokeelliseen dataan pohjautuvia seinäfunktioita, joilla suureiden seinämäkäyttäytymistä mallinetaan. Erilaisia k - ω -turbulenssimalleja käytettäessä *omegaWallFunction*- ja *kLowReWallFunction*-seinäfunktiot asettavat itsenäisesti seinänviereisen kontrollitilavuuden turbulenssiarvot oikein seinämäetäisyyden perusteella. Pyörreviskositeetin seinäfunktioille on useita vaihtoehtoja, mutta *nutUSpaldingWallFunction*-seinäfunktio on näistä helpoin

vaihtoehto, sillä käyttäjän ei tarvitse huolehtia siitä, missä kohdassa rajakerrosta ensimmäinen kontrollitulavuus on. (Liu 2017, s. 3–27) Turbulenssisuureet asetettiin sisääntulossa *turbulentIntensityKineticEnergyInlet*- ja *turbulentMixingLengthFrequencyInlet*-reunaehdoilla, joissa käytettiin turbulenssin intensiteettinä 5% sekä pyörteiden karakteristiseksi mitaksi valittiin elliptisen osuuden hydraulinen halkaisija kerrottuna kertoimella 0,07. Turbulenssiyhtälöissä esiintyvillä malliparametreille asetettiin taulukon 5.2 mukaiset Menter et al. (2003, s. 627) ilmoittamat arvot, joiden avulla muut yhtälöissä esiintyvät parametrit saadaan määritettyä laskennan aikana, esimerkiksi γ -parametri saadaan määritettyä yhteydestä $\gamma = \gamma_1 F_1 + \gamma_2 (1 - F_1)$. Lisäksi a_1 -kerroin perustuu Menterin (1993, s. 5) alkuperäisen mallin yhteydessä esitettyyn arvoon.

Taulukko 5.2 *k- ω SST -turbulenssimallin malliparametrit*

Vakio	σ_{k1}	σ_{k2}	$\sigma_{\omega1}$	$\sigma_{\omega2}$	β_1	β_2	β^*	$\gamma_{\omega1}$	$\gamma_{\omega2}$	a_1
Arvo	0,85	1,0	0,5	0,856	0,075	0,0828	0,09	0,555	0,44	0,31

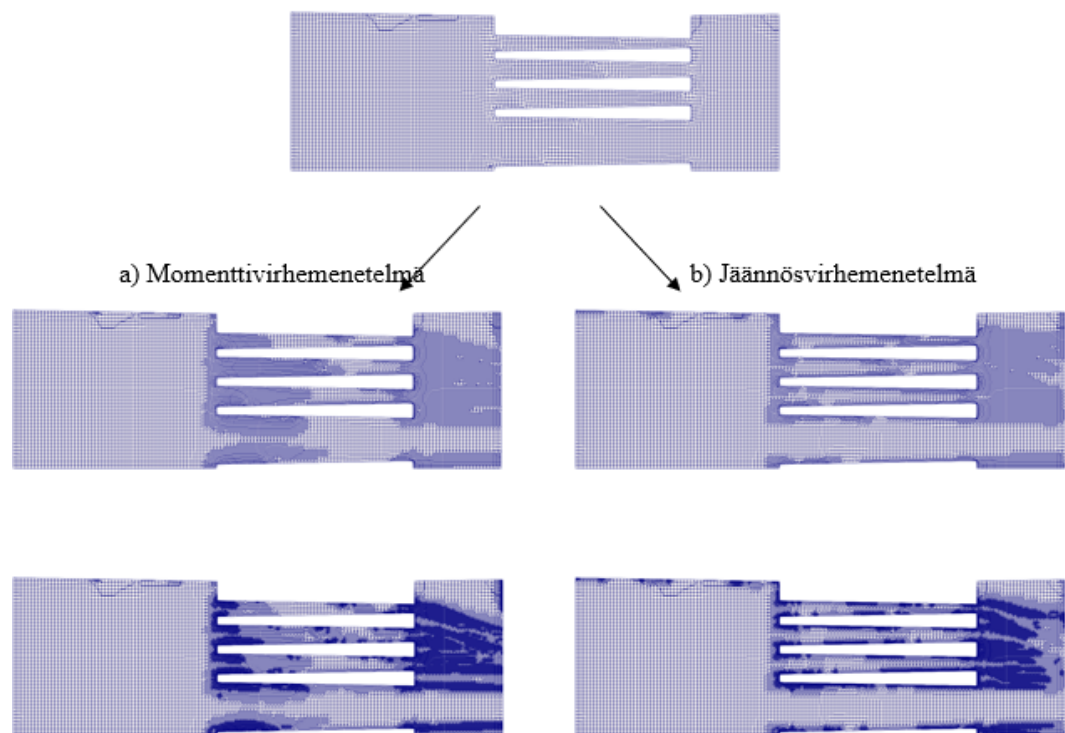
Laskentaa varten paineen alkuarvaukseksi annettiin arvo $p = 400 \text{ m}^2/\text{s}^2$, nopeusvektorin pituussuuntaiselle komponentille asetettiin $u_z = 20 \text{ m/s}$, turbulenssisuureille $k = 3,5 \text{ J/kg}$ ja $\omega = 40000 \text{ 1/s}$ sekä pyörreviskositeetille $\nu_t = 10^{-5} \text{ m}^2/\text{s}$. Koko laskennan ajaksi ilman aineominaisuuksiksi asetettiin 20°C lämpötilassa $\nu = 1,505 \cdot 10^{-5} \text{ m}^2/\text{s}$ ja $\rho = 1,188 \text{ kg/m}^3$, jotka perustuvat Millssin (2014, s. 920) taulukoimiin arvoihin. Virtaus-tilanteen Reynoldsin luku on sisäänmenon suureiden perusteella noin 22 000.

5.2 Menetelmän tulokset ja vertailu perinteisiin metodeihin

PimpleDyMFoam-ratkaisijassa tarvittavat raja-arvot muodostettiin Jasakin (1996, s. 228–232) suositusten mukaisesti, eli 1,5-kertaisesti diskretisointivirheen tilavuuskeskiarvon ylittävät laskentaverkon alueet tihennetään ja 0,5-kertaisesti alittavat harvennetaan. Alkuperäisessä lähteessä käytettiin diskretisointivirheen keskiarvoa, mutta tilavuuskeskiarvon havaittiin paremmin kuvaavan koko laskenta-alueen keskivirhettä. Tilavuuskeskiarvot määritettiin alkuarvaukseksi tehdyn laskentaverkon tuloksesta, josta saatiin keskiarvoiksi momenttinvirhe-estimaatilla 0,863 m/s ja jäännösvirhe-estimaatilla 0,280 m/s. Samalla asetettiin rajapintasolujen vähimmäismääräksi viisi kappaletta, että laskentaverkko muuttuu riittävän sileästi tihennettyjen alueiden välillä. Tämä oli hyvä kompromissi stabiiliuden kannalta, eikä tämä liioin kasvattanut adaptiivisten laskentaverkkojen kokoa.

Virhe-estimaattien muodostamisen jälkeen tehtiin kaksi testiajtoa, joista toisessa adaptiivisuutta ajoi momenttinvirhe-estimaatti ja toisessa jäännösvirhe-estimaatti. Analysointia helpottamaan geometrian keskeltä leikattiin kuvan 5.2 mukaiset siivut, joista voidaan nähdä selkeästi adaptiivisen algoritmin aikaansaannokset. Kahteen eri virhearvioon pohjautuvat adaptiiviset laskentaverkot ovat jonkin verran toisistaan poikkeavia.

Molemmilla adaptiivisilla iteraatioilla momenttivrhe-estimaatti pyrkii selkeästi tihentämään laskentaverkkoa rakojen alkupäässä enemmän kuin jäännösvirhe-estimaattia hyödyntävä algoritmi. Sen sijaan jäännösvirhepohjainen algoritmi keskittyy enemmänkin rajakerroksien tihentämiseen, eikä tihennä niin laaja-alaisesti rakojen alkuosuutta. Eroavaisuuksia voidaan havaita myös alueella, jossa virtaus poistuu suihkumaisesti raoista kohti laskenta-alueen ulostuloreunaa. Esimerkiksi ensimmäisen adaptiivisuusiteraation jälkeen molempien virhe-estimaattien vaikutuksesta pienempien suihkujen ulostuloalueella ja isomman suihkun reunamilla laskentaverkkoa on tihennetty lähes samalla tavalla, sillä näillä alueilla tiedettävästi aiheutuu suuria nopeusgradientteja. Toisella adaptiivisuuskierröksellä momenttivrhe-estimaatti pyrkii tihentämään näitä alueita vielä lisää, mutta jäännösvirhe-estimaatin tapauksessa laskentaverkkoa on tihennetty toisella kierroksella pääasiassa suihkujen välisellä alueella. Momenttivrhe-estimaatti tuotti ensimmäisellä iteraatiokierroksella kaiken kaikkiaan 2 130 000 kontrollitilavuuden laskentaverkon ja toisen kierroksen jälkeen kontrollitilavuuksia on lähes 7 050 000. Jäännösvirhe-estimaatin tuottamissa laskentaverkoissa on noin 2 010 000 ja 6 760 000 kontrollitilavuutta, eli molemmissa tapauksissa laskentaverkko kuta kuinkin kolminkertaistuu jokaisella adaptiivisella iteraatiokierroksella.

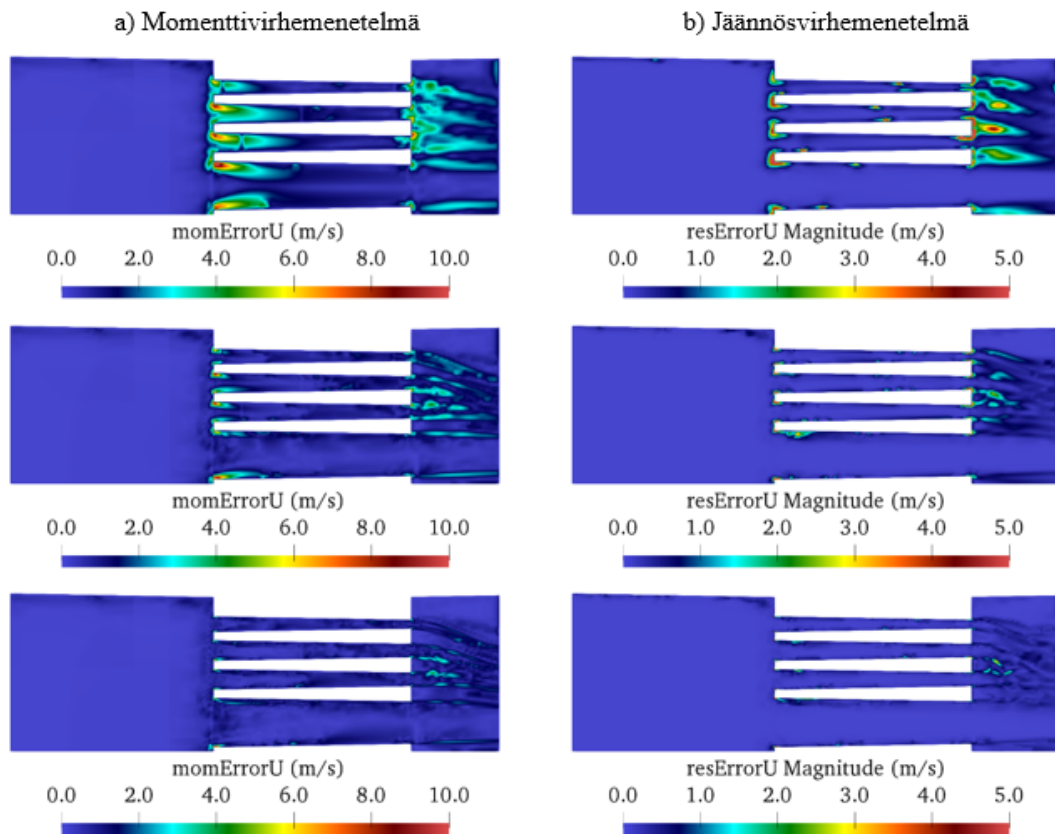


Kuva 5.2 Kahden eri virhe-estimaatin avulla luodut adaptiiviset laskentaverkot.

Edellä mainitut erot laskentaverkkojen välillä voidaan selittää virhe-estimaattien erilaisella toiminnalla. Kuvassa 5.3 esitettyjen momenttivrhe- ja jäännösvirhe-estimaattien jakaumat laskentaverkoissa poikkeavat toisistaan merkittävästi. Alkuperäisessä laskentaverkossa momenttivrhe-estimaatti ennustaa suurimpien diskretisointivirheiden olevan

rakovirtauksen alkupäässä, jossa virtaus kiihtyy ja samalla pyrkii irtoamaan seinämltä. Jäännösvirhe-estimaatti sen sijaan osoittaa, että suurimmat virheet aiheutuvat vain aivan rakovirtauksen alkupäässä ja myös suihkujen ulostuloalueiden välissä, sillä virtausnopeus muuttuu näillä alueilla merkittävästi. Jäännösvirhe-estimaatti löytää myös rakojen keski-osasta laskentaverkon virheitä aivan reunamien vierestä, jotka voidaan nähdä pistemäisinä virhealueina, mutta momenttinvirhe-estimaatti on epäherkempi näille virheille.

Molempien virhe-estimaattien tapauksessa voidaan kuvasta 5.3 todeta, että toteutettu h-adaptiivinen algoritmi on todella tehokas tapa vähentää diskretisointivirhettä. Molempien estimaattien ajava adaptiivisuus käytännössä poisti kokonaan virhekeskittymät analysoidulta alueelta, eikä merkittävästi tuonut lisävirhettä esimerkiksi huonontuneen laskentaverkon laadun vuoksi. Ensimmäisen adaptiivisuusiteraation jälkeen jäännösvirhe-estimaattia käytettäessä tosin aiheutuu hieman lisävirhettä isoimman raon alkuosaan, mutta virhe kuitenkin häviää seuraavassa adaptiivisuusiteraatiossa. Molemmissa tapauksissa laskentaverkkoihin jää rakojen sisääntuloreunalle laskentaverkon tihtämisestä huolimatta vielä vähän virhettä, sillä näissä uskotaan olevan mekaniikasta tutun säröongelman kaltainen singulariteetti, jonka suuruus ei välttämättä pienene, vaan sen koko pienenee.



Kuva 5.3 Analysoitujen laskentaverkkojen virhe-estimaatit

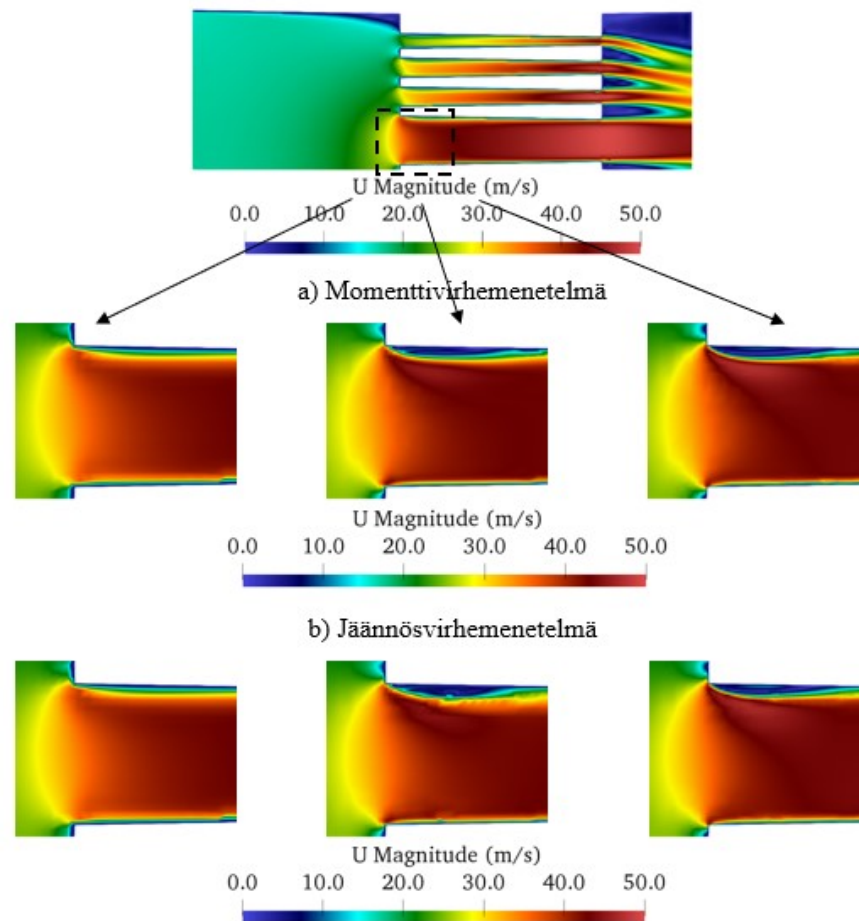
Momentti- ja jäännösvirhejakaumia analysoitaessa voidaan huomata, että estimaattien suuruus poikkeaa varsin paljon toisistaan, sillä jäännösvirheen suuruus on vain noin puolet tai kolmasosa momenttinvirheen ennustamasta diskretisointivirheestä. Momenttinvirhe

on myös huomattavasti suuremmalle alueelle jakautunut, sillä jäännösvirhe on hyvin lo-kaalia. Virhe-estimaattien eroavaisuus on havaittu myös kirjallisuudessa (Jasak 1996).

Molempien virhe-estimaattien tunnetaan tuottavan lähes samanlaisia tuloksia 2D-malli-ongelmissa ja arvioivan todellista diskretisointivirheen tasoa varsin tarkasti, eikä näiden toiminta merkittävästi kärsi laskentaverkon topologisista virheistä. Momenttivirhe-esti-maatin tunnetaan kuitenkin jonkin verran yliarvioivan diskretisointivirhettä, kun lasken-taverkkoa tihennetään. Virhe-estimaattien toiminta kärsii silmäänpistävästi vasta siinä vaiheessa, kun numeerinen ratkaisu sisältää poikkeuksellisen paljon numeerista dif-fuusiota, eli kyseessä on virhe-estimaattien kannalta äärimmäinen rasisuskoe. Jäännös-virhe-estimaatin voidaan tällöin todeta arvioivan todellisen diskretisointivirheen maksi-mia kohtuullisesti, mutta momenttivirheen ennustama maksimivirhe poikkeaa todelli-sesta arvosta jo huomattavasti. Keskivirheen tapauksessa päinvastainen on havaittu pitä-vän paikkaansa, eli momenttivirhe-estimaatti arvioi keskivirheen suuruutta kohtuullisesti, mutta jäännösvirhe-estimaatti merkittävästi aliarvioi keskivirhettä. Kyseisellä tavalla saa-dut havainnot eivät välttämättä kuvaa todellisia laskentaverkkoja erityisen hyvin, vaan kyseessä on pikemminkin se, miten hyvin virhe-estimaatit toimivat äärimmäisissä tapauk-sissa. Realistisemmassa virtausmekaniikan 2D-sovelluksessa jäännösvirhe-estimaatti ky-kenee arvioimaan sekä keskivirheen että maksimivirheen hyvin tarkasti pienesti yliarvi-oiden näiden suuruutta, mutta momenttivirhe-estimaatin on todettu yliarvioivan sekä kes-kivirhettä että maksimaalista virhettä varsin reilusti. Poikkeuksellisen tiheillä laskenta-verkoilla virhe-estimaattien toiminta saattaa muuttua hieman. (Jasak 1996, s. 201–217) Edellä mainittujen havaintojen vuoksi jäännösvirhe-estimaatin voidaan sanoa olevan pa-rempi osoittamaan diskretisointivirheen suuruutta tutkitussa testigeometriassa, vaikka virhe-estimaattien tarkkaa käyttäytymistä 3D-sovelluskohteissa ei tunnetaakaan. Molem-mat virhe-estimaatit antavat ehdottomasti käyttökelpoisen arvion diskretisointivirheen ylärajasta, eikä pienillä eroavaisuuksilla välttämättä ole niin suurta merkitystä, jos halu-taan poistaa karkeimmat virheet numeerisesta ratkaisusta, sillä molemmat virhe-estimaatit tihentävät laskentaverkkoa lähes samalla tavalla.

Liiketyhtälöiden diskretisointivirheen pienentyminen ei ole vain teoreettinen hyöty, vaan tällä on vaikutuksia myös nopeuskenttään, erityisesti rakojen alkupäähän, kuten kuvasta 5.4 nähdään. Alun perin virtaus pysyy kiinni seinämissä, mutta adaptiivisuuden ansiosta liikeyhtälöiden ratkaisua on tällä alueella saatu huomattavasti parannettua, jolloin virtaus todellisuudessa irtoaakin sisääntuloreunalla ja vasta myöhemmin kiinnittyy taas raon sei-nämiin. Samalla myös sisääntuloalueen nopeuskenttä muuttuu sileämmäksi. Jäännös-virhe-estimaattia käytettäessä ensimmäisen adaptiivisen iteraation jälkeen on nähtävissä jonkin verran adaptiivisuuden aikaansaamaa diskretisointivirhettä, sillä virtauksen irtau-tumiskohdassa kenttä ei ole täysin sileä vaan kontrollitulavuuksien rajat ovat jonkin verran näkyvissä ratkaisussa. Toinen adaptiivisuuskierron palauttaa taas sileän nopeuskentän, eikä eri virhe-estimateilla saaduissa lopputuloksissa ole käytännössä lainkaan eroa.

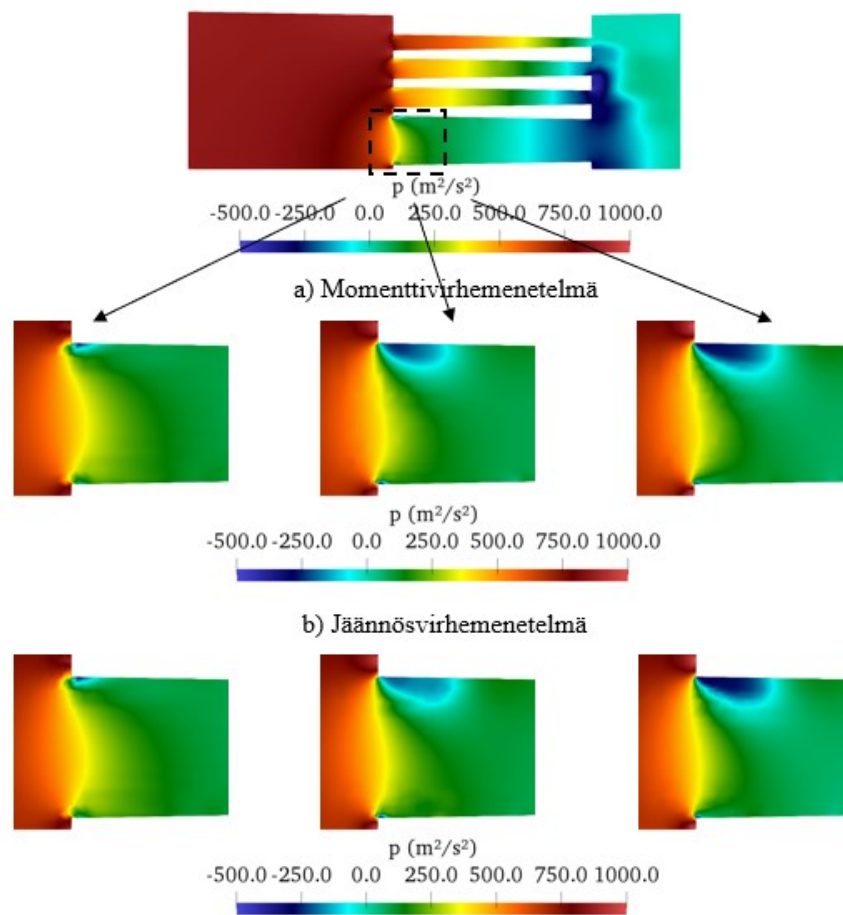
Kuvassa 5.5 esitetyissä painekentän muutoksissa on havaittavissa samanlaista käyttäytymistä kuin nopeuskentässä, sillä liikeyhtälöiden tarkempi ratkaisu parantaa myös painekentän ratkaisua. Alkuun painekenttä vaikuttaa varsin sileältä, mutta adaptiivisuudella saadaan aikaan paljon lisäinformaatiota. Adaptiivisissa ratkaisuisa esille nousee uusi ja varsin laaja matalapainealue, joka on käytännössä olematon alkuperäisessä tuloksessa. Virhe-estimaattien välillä ei ole suurta eroa, mutta momenttinvirhe-estimaatin tuottama painekenttä on hieman sileämpi.



Kuva 5.4 Nopeuskentän muuttuminen adaptiivisuuden edetessä

Seuraavaksi adaptiivisuuden suorituskykyä vertailtiin perinteiseen laskentaverkon mallinnusmenetelmään, jossa mallinnetaan rajakerrokset tarkasti siten, että seinämän viereisen kontrollitilavuuden keskipiste sijaitsee keskimäärin viskoosin rajakerroksen sisällä, eli rajakerrosten ratkaisua voidaan tällöin pitää asianmukaisena. Lisäksi koko rako-osuus tihennettiin vielä erikseen, jotta tämän sisältämät virtausilmiöt saadaan vielä paremmin esille. Lopputulema voisi hyvin kuvata asiantuntijan tekemää laskentaverkkoa tutkittavalle sovelluskohteelle ja tämän jälkeen generoitiin kaksi vastaavaa laskentaverkkoa, mutta tiheämmällä koolla. Lisäksi mielenkiinnon vuoksi näille laskentaverkoille käytettiin perinteistä Greenin–Gaussin gradienttimenetelmää, mutta muita konvektiomenetelmiä ei kokeiltu, sillä Gamma-menetelmän uskottiin antavan hyviä tuloksia sellaisenaan.

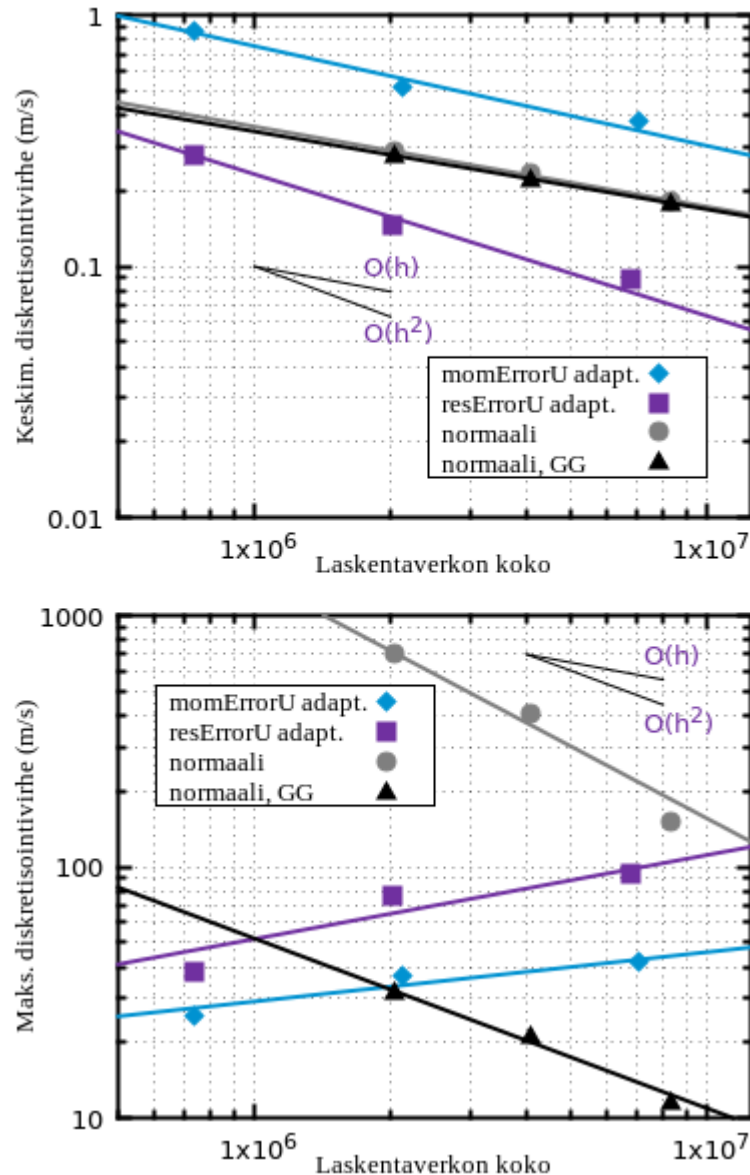
Adaptiivisuuden erinomaista suorituskykyä havainnollistaa kuva 5.6, jossa logaritmisella asteikolla esitetyt tulokset on muodostettu jäännösvirhe-estimaatin avulla, ja momenttivrhepohjaisille adaptiivisille laskentaverkoille on käytetty poikkeuksellisesti momenttivrhe-estimaattia. Diskretisointivirheen tilavuuskeskiarvon pienentämisessä adaptiiviset menetelmät kykenevät lähes odotettuun toisen kertaluvun suorituskykyyn, mutta perinteisten mallien tulokset ovat vain noin ensimmäistä kertalukua, eikä gradienttimenetelmän vaihto muuta tilannetta. Ilmiön uskotaan selittyvän sillä, että Gamma-menetelmä käyttää adaptiivisissa laskentaverkoissa huomattavasti enemmän toisen kertaluvun konvektiomenetelmiä, kun taas perinteisissä laskentaverkoissa joudutaan painottamaan enemmän ensimmäisen kertaluvun konvektiomenetelmää, jolloin myös tarkkuus kärsii.



Kuva 5.5 Painekentän paraneminen adaptiivisuuden vaikutuksesta

Diskretisointivirheen maksimi-arvon kohdalla voidaan tehdä mielenkiintoisia havaintoja. Adaptiivisuutta käytettäessä algoritmi generoi satunnaisesti paikkoihin topologisesti huonolaatuisempia kontrollitilavuuksia, joiden vaikutuksesta erityisesti singulariteettipisteissä diskretisointivirhe kasvaa. Rajakerrosten mallintaminen luo kaikkiin perinteisiin laskentaverkkoihin samanlaisia topologisia virheitä rakojen sisään- ja ulostuloreunoille, jotka aiheuttavat suuren maksimivirheen erityisesti pienimmän neliösumman gradienttimenetelmää käytettäessä. Tasainen tihentäminen kuitenkin poistaa näiden topologisten virheiden määrää, mutta näin ei käy adaptiivisissa laskentaverkoissa. Pienimmän

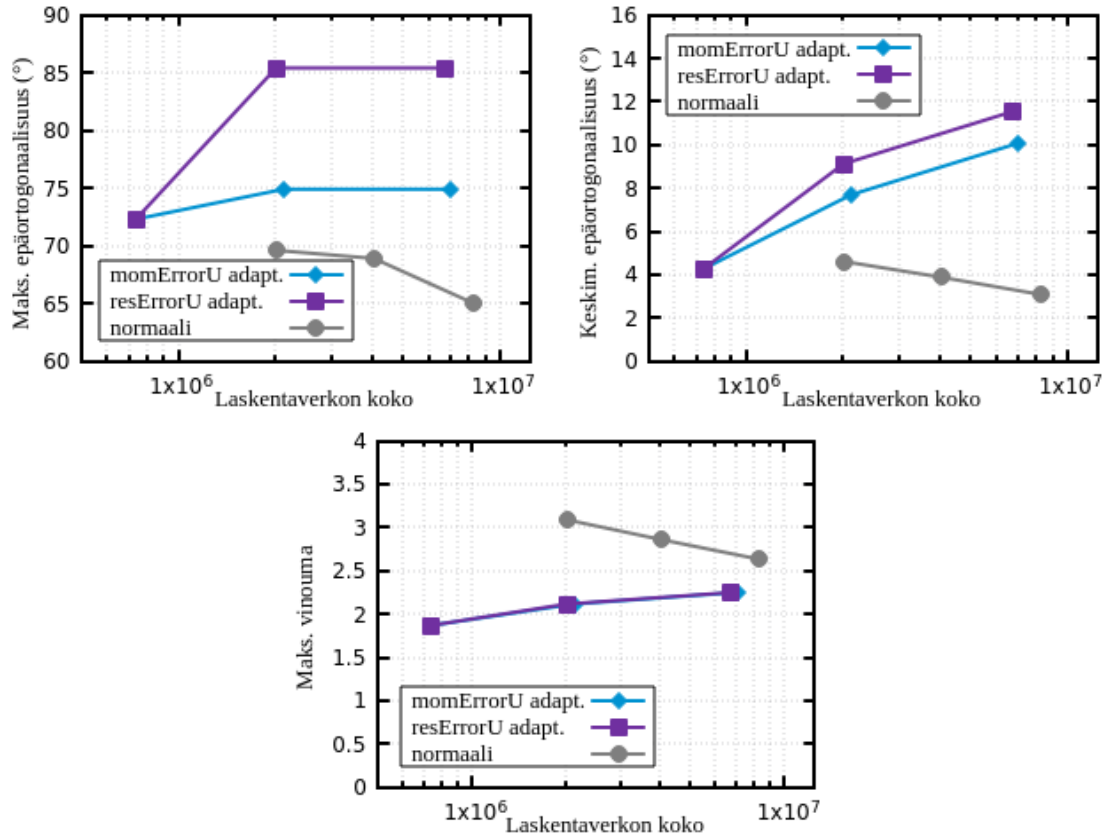
neliösumman menetelmä vaikuttaa olevan varsin herkkä singulariteetipisteille, sillä maksimivirhe on noin kertalukua suurempi kuin Greenin–Gaussin menetelmän tapauksessa. Poikkeuksellisen suuri maksimivirheen pienenemisnopeus viittaa myös singulariteetin vaikutukseen. Greenin–Gaussin gradientti on selkeästi epäherkempi singulariteeteille, koska maksivirheen suuruus on samaa kertaluokkaa kuin adaptiivisilla laskentaverkoilla.



Kuva 5.6 Tulosten diskretisointivirheiden tilavuuskeskiarvot ja maksimi-arvot

Kuvaan 5.7 on koottu OpenFOAM:in *checkMesh*-työkalun avulla laskettuja erilaisia ja keskeisiä laskentaverkon laadullisia ominaisuuksia, jotka aiheuttavat edellä mainitun maksimivirheen kasvamisen adaptiivisissa laskentaverkoissa. Topologiavirheet kasvavat, sillä kontrollitilavuuksien pilkkominen aina kahdeksaan osaan nostaa laskentaverkon epäortogonaalisuutta jokaisella adaptiivisella iteraatiolla. Myös maksimaalinen vinouma kasvaa hieman. Momenttivirhe-estimaatin käyttö on hieman hellävaraisempi laskentaverkon virheiden kannalta. Manuaalisesti tehtyjen laskentaverkkojen käyttäytyminen on

päinvastainen, sillä näiden epäortogonaalisuus ja vinouma vähenee, kun laskentaverkkoa tihennetään. Vinouman kannalta adaptiiviset laskentaverkot ovat poikkeuksellisesti laadultaan parempia, sillä rajakerrosten lisääminen laskentaverkkoon aiheuttaa niihin luonnostaan enemmän vinoumaa.



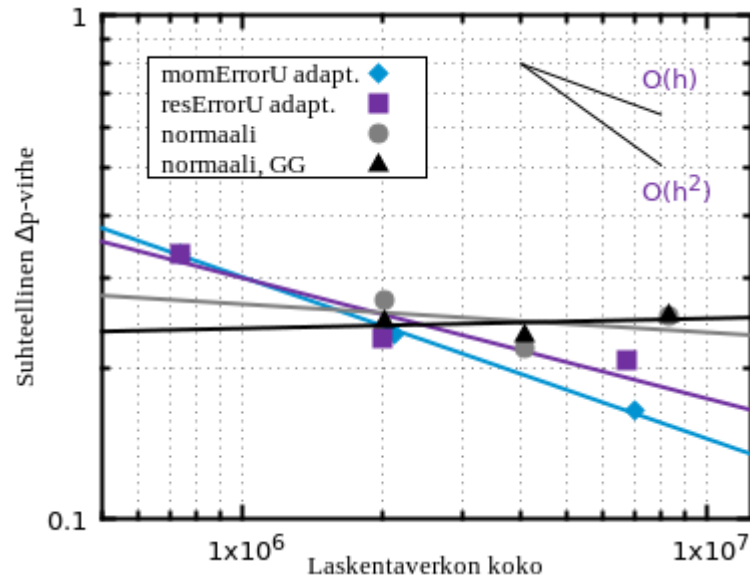
Kuva 5.7 Ratkaistujen laskentaverkkojen epäortogonaalisuus ja vinouma

Laskentaverkon heikentyneellä laadulla on omat vaikutuksensa myös laskennan etenemiseen. Hyviä laskenta-asetuksia haettaessa huomattiin epäortogonaalisuuden aiheuttama haaste, sillä laskennan suppeneminen oli varsin riippuvainen epäortogonaalisen lisäkorjauksen määrästä. Jos laskennassa sallittiin täysi epäortogonaalisuuskorjaus, laskenta usein divergoi adaptiivisilla laskentaverkoilla. Sen sijaan normaalit laskentaverkot eivät olleet niin herkkiä kyseiselle asetukselle, koska näiden epäortogonaalisuus oli huomattavasti pienempi, eikä vinouman vaikutusta suoranaisesti huomattu laskennan aikana.

5.3 Kokeellisen datan ja virhearvioiden vertailu

Viimeiseksi adaptiivisen algoritmin suorituskykyä tutkittiin vertaamalla numeerisesta ratkaisusta haetun painegradientin arvoa kokeelliseen dataan, sillä tutkimuskohteena olevasta virtausmittarista tunnetaan painekammioiden välinen paine-ero. Virtausmittarissa ensimmäinen painekammio mittaa noin puolivälistä rakovirtauksen staattista painetta ja toinen lähes rakovirtauksen lopusta. Kuvaan 5.8 on laskettu numeeristen ratkaisujen suhteellinen virhe, kun kokeellista dataa pidetään täysin tarkkana referenssisuureena. Tähän

liittyy tietysti vielä oma epävarmuutensa, mutta tähän ei työn puitteissa voida vaikuttaa. Painereunaehto- ja virtausmittarin läpi menevä tilavuusvirta saattaa hieman vaihdella ratkaisujen välillä, mutta tämä otettiin erikseen huomioon, koska saatavilla oli neljännen asteen polynomifunktio mitatulle paine-erolle tilavuusvirran funktiona.

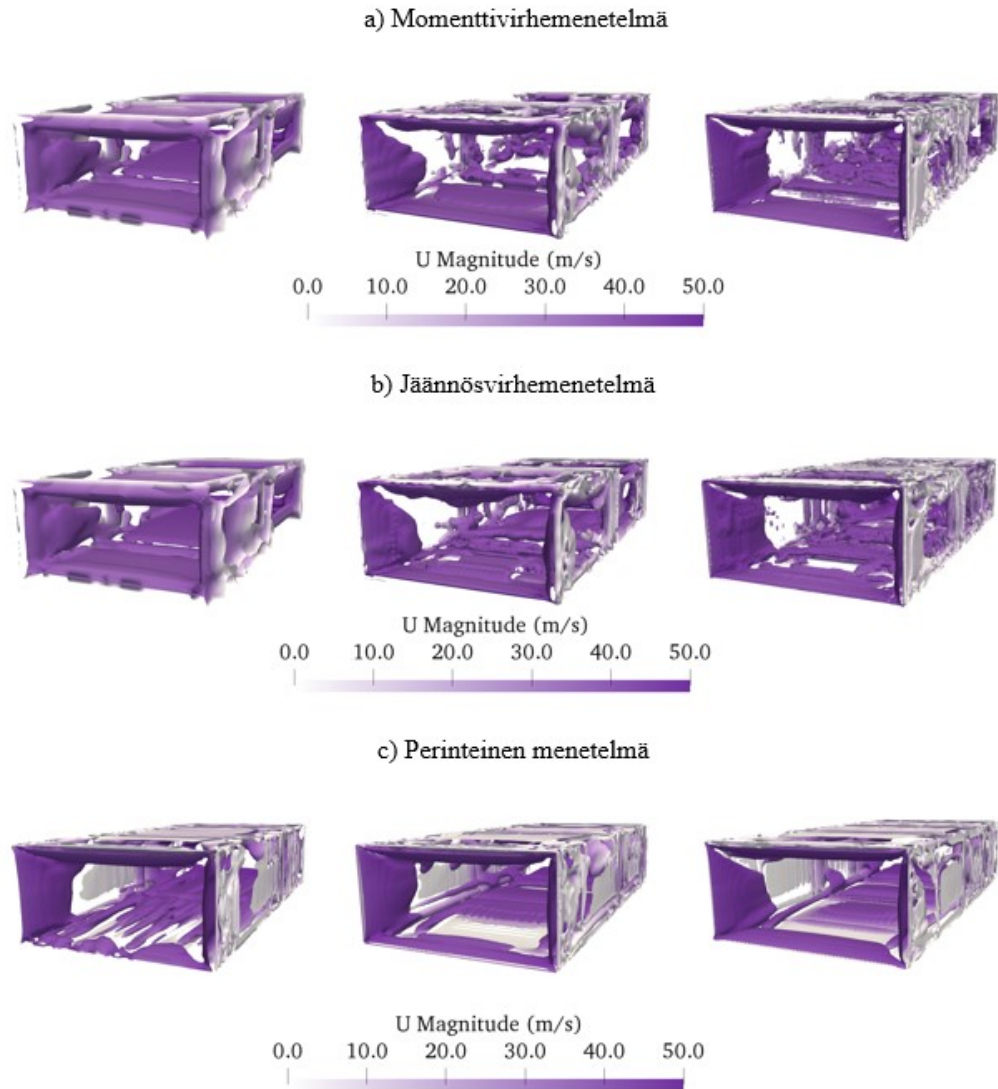


Kuva 5.8 Painegradientin suhteellinen virhe

Kuvasta 5.8 nähdään, että momenttivirhe-estimaattiin perustuva adaptiivisuus tuottaa ylivoimaisesti parhaita tuloksia painegradientin kannalta, mutta myös jäännösvirhe-estimaattiin pohjautuvat tulokset ovat varsin hyviä. Perinteisellä menetelmällä saadut tulokset ovat hieman poikkeuksellisia, sillä painegradientin suhteellinen virhe ei pienene monotonisesti, eikä molemmilla gradienttimenetelmillä kaikista tihein laskentaverkko enää paranna ratkaisun tulosta, vaan pikemminkin heikentää sitä. Aiemmin kuitenkin todettiin kaikkien ratkaisujen diskretisointivirheen pienenevän. Greenin–Gaussin gradienttimenetelmän käyttäytymistä voidaan odottaa teorian perusteella, mutta pienimmän neliösumman menetelmä vaikuttaa käyttäytyvän samalla tavalla, vaikka gradienttien diskretisointi pitäisi olla vähintään ensimmäistä kertalukua. Adaptiiviset tulokset eivät osoita painegradientin suppenevan ensimmäisen kertaluvun nopeudella, vaan tästä jäädyään hieman, mutta tämä voidaan osittain selittää sillä, että laskennan yhteydessä jouduttiin rajoittamaan epäortogonaalisen lisäkorjauksen suuruutta. Tämä selittäisi myös sen, miksi jäännösvirhe-estimattia käytettäessä jäädyään hieman momenttivirhepohjaisen adaptiivisuuden suorituskyvystä, sillä näissä laskentaverkoissa maksimaalinen epäortogonaalisuus oli selkeästi suurempi. Kuitenkin adaptiivisten laskentaverkojen laatu on huonompi ja nämä silti kykenevät monotonisesti vähentämään painegradientin suhteellista virhettä, joten perinteisille menetelmille havaittu ilmiö ei voi johtua pelkästään topologisista virheistä.

Kaikki saadut tulokset aliarvioivat todellista painegradienttia jonkin verran. Tarkemmissa tuloksissa esiintyy selkeästi enemmän turbulenssia, jolloin myös rako-osuuden yli laskeutu painehäviö pyrkii kasvamaan. Tätä turbulenssia voidaan havainnollistaa esimerkiksi

yleisesti virtausmekaniikassa Q-kriteerinä tunnetulla nopeusgradientin toisella invariantilla (Nieuwstadt et al. 2016, s. 154–155), joten kuvaan 5.9 on havainnollistettu Q-kriteerin 10^5 1/s^2 tasa-arvopintoja testigeometrian suurimmasta kanavasta. Kuvassa tasa-arvopinnat on värjätty nopeusvektorin suuruudella.



Kuva 5.9 Rakovirtauksen turbulenssin kasvaminen eri menetelmillä

Turbulenssin lisääntyminen adaptiivisuutta käytettäessä on hyvin selvä. Turbulenssi muodostuu nopeasti kolmiulotteiseksi, mutta perinteisillä menetelmillä turbulenssi on pikemminkin pitkittäissuuntaista. Kyseisten laskentaverkkojen sisältämät kontrollitilavuudet ovat seinämällä niin ohuita niiden pituuteen nähden, että ne eivät kykene hyvin mallintamaan heti rakovirtauksen alussa olevia suuria nopeusgradientteja, koska nopeusgradientti ei siellä ole pelkästään seinämän normaalin suuntainen vaan myös virtaussuuntainen. Tällöin myös painegradientti muuttuu, eikä tätä välttämättä saada laskettua täysin oikein. Tiheimmän perinteisen laskentaverkon tulos pitäisi olla huomattavan tarkka mutta näin ei kuitenkaan käynyt, vaikka sen keskimääräinen dimensioton seinämäetäisyys on noin 1,2, eli seinämän viereiset kontrollitilavuudet ovat reilusti viskoosin rajakerroksen sisällä.

6. YHTEENVETO

Ensimmäistä tutkimuskysymystä tutkittaessa, eli miten kontrollitulavuusmenetelmällä päästään laskentaverkon tyypistä riippumatta toisen kertaluvun tarkkuuteen, huomattiin kyseisen vaatimuksen olevan pääasiassa kiinni vain valituista diskretisointimenetelmistä ja laskentaverkon topologisista virheistä. Tästä syystä työssä käytiin läpi pääpiirteittään virtausmekaniikassa käytettävän kontrollitulavuusmenetelmän erilaisia virhelähteitä, ja huomattiin, että tavoiteltuun toisen kertaluvun tarkkuuteen ei aina päästä insinöörisovelluksissa, joissa laskentaverkko on lähes poikkeuksetta epäsäännöllinen. Näin ollen työssä tutustuttiin tarkemmin erilaisiin diffuusio-, gradientti- ja konvektiomenetelmien virhelähteisiin sekä näiden toimivuuteen epäsäännöllisissä laskentaverkoissa ja samalla sivuttiin erilaisia aikatermien ja lähde-termien asianmukaisia diskretisointitapoja.

Työssä havaittiin diffuusiotermien diskretisoinnin toisen kertaluvun tarkkuudella olevan varsin suoraviivaista niiden elliptisten ominaisuuksien vuoksi, sillä keskeisdifferenssimenetelmä tuottaa erinomaisia tuloksia kyseisille termeille. Laskentaverkon epäortogonaalisuus asettaa oman lisähaasteensa, sillä tämän huomioonottamisessa on useita vaihtoehtoja. Alirelaksoitu epäortogonaalisuuskorjaus osoittautui parhaaksi vaihtoehdoksi, erityisesti niissä tapauksissa, joissa laskentaverkon epäortogonaalisuus on suuri. Epäortogonaalisuuden korjaus on riippuvainen myös gradienttimenetelmistä, joten näillekin jouduttiin etsimään mahdollisimman hyvät diskretisointimenetelmät.

Gradienttimenetelmistä parhaimmaksi todettiin painokertoimiin pohjautuva pienimmän neliösumman menetelmä, koska yleisesti käytetyn Greenin–Gaussin gradienttimenetelmän hyödyntäminen insinöörisovelluksissa voi pahimmassa tapauksessa johtaa tarkkuudeltaan jopa nollannen kertaluvun gradientteihin. Pienimmän neliösumman menetelmän todettiin olevan vähintään ensimmäistä kertalukua vastaavissa tapauksissa. Gradienttien määrittäminen toisen kertaluvun tarkkuudella on edelleenkin merkittävän haastavaa kontrollitulavuusmenetelmällä, sillä toisen kertaluvun tarkkuuden havaittiin toteutuvan vain hyvin rajoitetuissa mallitapauksissa työssä tutkituille gradienttimenetelmille.

Konvektiomenetelmiin jouduttiin käyttämään enemmän aikaa, sillä konvektiotermien diskretisoinnin lopputuloksen huomattiin olevan varsin kehno keskeisdifferenssimenetelmällä, sillä konvektio on hyvin virtaussuunnasta riippuvaa. Keskeisdifferenssimenetelmän käyttö vaatisi erittäin tiheän laskentaverkon, jotta menetelmän stabiiliusehto saataisiin täytettyä. Tästä syystä tutustuttiin erilaisiin vaihtoehtoisiin ensimmäisen ja toisen kertaluvun konvektiomenetelmiin, ja näiden toiminta havaittiin kohtuulliseksi. Merkittäväksi heikkoudeksi perinteisille toisen kertaluvun konvektiomenetelmille todettiin niiden epäfysikaalisuus, joten tästä syystä työssä jouduttiin paneutumaan TVD- ja NVD-viitekehyksiin, josta jälkimmäisessä johdettu Gamma-menetelmä osoittautui parhaaksi vaihtoehdoksi konvektiotermien diskretisointiin ja kyseisen menetelmän todettiin tuottavan

fysikaalisia tuloksia myös epäsäännöllisissä laskentaverkoissa, eikä menetelmä lisää merkittävästi numeerista diffuusiota simulaatiotuloksiin.

Yhtälöissä esiintyvien lähdetermien osalta kontrollitulavuusmenetelmä on yksinkertainen, mutta numeerista ratkaisuprosessia voidaan stabiloida, jos lähdeterminit linearisoidaan sopivasti. Aikatermien diskretisointiin ei paneuduttu syvällisemmin, vaan näistä nostettiin pikaisesti esille eri menetelmien stabiilius ja näiden tarkkuus.

Työn alkupuolella asetettiin toinen kontrollitulavuusmenetelmän tarkkuuteen liittyvä tutkimuskysymys, eli miten työssä ratkaistaville RANS-yhtälöille saadaan muodostettua hyviä virhearvioita ja mielellään sellaisia, jotka luotettavasti arvioivat diskretisointivirheen ylärajaa. Vastausta etsittäessä käännettiin katse elementtimenetelmän puolelle johdettuun virhearvioiden teoriapohjaan ja tämän virhearvioteorian todettiin olevan sovellettavissa myös kontrollitulavuusmenetelmään, jossa virhearvioiden käyttö ei ole arkipäivää. Virhearviotyyppejä huomattiin olevan varsin monta erilaista, kuten gradienttivrheisiin pohjautuvat virhearviot, sekä erilaiset laskennan jälkeen tehtävät eksplisiittiset ja implisiittiset virhearviot. Samalla paneuduttiin duaalimenetelmälle johdettuihin virhearvioihin pikaisesti. Ongelmaksi nousi, että virhe-estimaateissa käytettävät jäännökset ja gradienttihyppäykset eivät ole suoraan virtauslaskennan tuloksista saatavilla. Täten lähdettiin hakemaan erityisesti kontrollitulavuusmenetelmään johdettuja virhearviomenetelmiä.

Kontrollitulavuusmenetelmään johdettuja virhearviomenetelmiä löydettiin työssä useita, kuten Richardsonin ekstrapolointiin pohjautuvia menetelmiä, momenttivrhemenetelmä sekä jäännösvirhemenetelmä. Näistä momenttivrhe- ja jäännösvirhemenetelmän todettiin soveltuvan erittäin hyvin adaptiiviseen laskentaverkon hienontamiseen, koska näiden tuottama virhe-estimaatti oli kohtuullisen helppo toteuttaa myös käytännössä. Näiden yhteydessä käytiin läpi myös elementtimenetelmän puolelta lainattu, implisiittinen tasapainotettujen jäännösten menetelmä, jolla saadaan arvioitua numeerisen ratkaisun diskretisointivirheen energianormia erinomaisella tarkkuudella. Aikaisempi elementtimenetelmän virhearvioteoriaan paneutuminen osoittautui siis hyödylliseksi. Työssä päädyttiin kuitenkin toteuttamaan eksplisiittiset momenttivrhe- ja jäännösvirhe-estimaatit, ja näiden suorituskyky osoittautuikin myöhemmin erittäin hyväksi. Virhearvioiden lopputulos on hieman erilainen elementtimenetelmän virhearvioihin nähden, sillä valitut virhearviot tuottavat joko skalaari- tai vektorimuotoisen virhearvion, eikä saatu virhearvio täten vastaa elementtimenetelmän puolella käytettyä diskretisointivirheen energianormia. Virhe-estimaattien muoto on kuitenkin ehdottomasti havainnollisempi, sillä RANS-yhtälöihin sovellettaessa virhe-estimaattien yksikkö vastaa kätevästi nopeusvektorin yksikköä.

Virhearviomenetelmien käytännön toteutukseen valittiin OpenFOAM v5.0 -ohjelmisto, johon löydettiin osittain valmiita lisäosia foam-extend 4.0 -työkalupakista, joilla tarvittavat virhearviot saatiin toteutettua. Ohjelmoinnin työmäärä oli näin vielä hallittavissa työn aikataulun puitteissa. Valmiita kirjastoja ja työkaluja saatiin hyödynnettyä siten, että vain noin puolet virhearvioiden sisältämästä koodista jouduttiin toteuttamaan alusta lähtien.

Adaptiiviset ominaisuudet saatiin kytkettyä päälle ilman sen kummempaa lisätyötä, sillä nämä ominaisuudet löytyivät ohjelmistosta jo entuudestaan. Adaptiivinen laskentaverkko saatiin muodostettua jakamalla aina jokainen kontrollitilavuus pienempiin osiin, jos tämän diskretisointivirhe oli liian suuri laskentaverkon tilavuuskeskiarvoon nähden ja laskentaverkkoa harvennettiin siellä, missä diskretisointivirhe oli huomattavasti alempi.

Tämän jälkeen työssä keskityttiin tutkimaan momenttivirhe- ja jäännösvirhepohjaisen adaptiivisuuden suorituskykyä, joka todettiin erittäin hyväksi. Diskretisointivirheen tilavuuskeskiarvon havaittiin pienenevän lähes toisen kertaluvun nopeudella. Adaptiivisuuden tehokkuutta haluttiin verrata myös perinteiseen mallinnusmenetelmään, jossa laskentaverkko tehtiin seinämällä niin tiheäksi, että rajakerroksissa virtauskentän ratkaisu on hyvä, ja muualle mielenkiinnon kohteena oleville alueille lisättiin manuaalisesti lisäresoluutiota. Lopputuloksena olevien laskentaverkkojen huomattiin pienentävän diskretisointivirhettä vain noin ensimmäisen kertaluvun nopeudella, eli adaptiivisen algoritmin suorituskyvystä jäädään huomattavasti. Erityispiirteinä adaptiivisissa laskentaverkoissa on se, että adaptiivinen algoritmi kykenee tuomaan uusia virtausilmiöitä simulaatitulosiin, kuten esimerkiksi virtauksen irtoamista seinästä ja kokonaan uusia matalapainealueita.

Taulukko 6.1 Perinteisten mallinnusmenetelmien ennustetut laskentaverkon koot ja laskenta-ajat verrattuna jäännösvirhepohjaiseen adaptiivisuuteen

Menetelmä	Adaptiivinen (jäännösvirhe)	Normaali	Normaali (Green-Gauss)
Laskentaverkon koko (miljoonaa kpl)	6,8	81,9	80,4
Laskenta-aika (h)	6,0	47,3	39,7

Aiemmin laskettujen diskretisointivirheiden tilavuuskeskiarvojen avulla voidaan ekstrapoloida perinteisten mallinnusmetodien vaatima laskentaverkon koko, jolla saavutettaisiin yhtä tarkka lopputulos kuin kaikista tarkin jäännösvirhepohjaisella adaptiivisella algoritmilla saatu tulos. Olettamalla laskenta-ajan riippuvan lineaarisesti laskentaverkon koosta, voidaan saada taulukon 6.1 mukaisia tuloksia. Laskentaverkkojen parantunut hyötysuhde näkyy selkeästi, sillä yhtä tarkkoihin tuloksiin vaadittaisiin jopa 10-kertaisesti suuremmat laskentaverkot perinteisellä mallinnusmenetelmällä, ja Greenin–Gaussin gradienttimenetelmän käyttäminen vaikuttaa hieman suotuisammalta rajakerroksia mallintavissa laskentaverkoissa. Laskenta-aikakin nousisi perinteisillä laskentaverkoilla vähintään 6,6-kertaiseksi adaptiivisen ajon kumulatiiviseen laskenta-aikaan nähden, joten adaptiivisuuden tuomat edut ovat kiistattomat.

Adaptiivisen algoritmin havaittiin tuottavan laskentaverkkoon varsin paljon topologisia virheitä, jotka aiheuttivat tarpeen sille, että epäortogonaalisen lisäkorjauksen suuruutta joudutaan rajoittamaan. Perinteisissä laskentaverkoissa topologia paranee sitä mukaa,

kun laskentaverkkoa tihennetään, eikä näille välttämättä olisi tarvittu kyseistä lisärajoitusta tehdä, mutta näin päädyttiin tekemään, että tuloksissa käytetyt asetukset olivat keskenään samoja. Lisääntyneistä laskentaverkon topologiavirheistä huolimatta adaptiivinen algoritmi kuitenkin kykenee pienentämään diskretisointivirhettä erinomaisesti. Vaikka adaptiivista algoritmia kyettiin käyttämään laskentatehon ja muistirajoitteiden puitteissa vain kahteen kertaan, saatiin kuitenkin vertailukelpoisia tuloksia perinteisiin mallinnusmenetelmiin nähden. Jälkikäteen ajateltuna adaptiivisessa algoritmista käytettäviä raja-arvoja olisi voinut säätää siten, että olisi saatu useampi määrä adaptiivisia iteraatioita, mutta tällöin olisi jouduttu poikkeamaan kirjallisuudessa suositelluista raja-arvoista.

Lopuksi työssä verrattiin adaptiivisten ja perinteisten laskentaverkkojen ennustamaa painegradienttia kokeellisiin tuloksiin nähden. Havaittiin, että adaptiiviset laskentaverkot jäävät hieman odotetusta ensimmäisen kertaluvun suorituskyvystä, sillä erilaisten keino-tekoisten stabilointimenetelmien käyttö epäortogonaalisen lisäkorjauksen yhteydessä uskottiin vähentävän gradienttien tarkkuutta. Perinteisten laskentaverkkojen tapauksessa painegradientin virheen vähenemisen ei havaittu olevan monotonista gradienttimenetelmästä riippumatta, mutta pienimmän neliösumman menetelmän havaittiin olevan hieman parempi perinteisten laskentaverkkojen tapauksessa.

Erityisesti painegradientin laskennassa huomattiin se, että numeeriset laskentatulokset perinteisillä laskentaverkoilla saattavat antaa harhaanjohtavia tuloksia, sillä lisähienontaminen ei välttämättä enää muuta mielenkiinnon kohteena olevaa suuretta merkittävästi, eikä suure välttämättä suppene oikeaa arvoa kohti. Karkeasti ottaen voidaan todeta, että jos liikeyhtälöiden keskivirhe pienenee toisen kertaluvun nopeudella, voidaan muidenkin suureiden, kuten esimerkiksi mitatun painegradientin virheen, olettaa pienevän lähes teorian mukaisella miniminopeudella. Sama voitane nostaa esille myös perinteisille laskentaverkoille, sillä liikeyhtälöiden ratkaisujen ollessa tarkkuudeltaan ensimmäistä kertalukua, voitane odottaa painegradientin olevan määritetty pahimmillaan vain nollannen kertaluvun tarkkuudella. Virhe-estimaattien ja kokeellisen datan voidaan siis sanoa korreloivan kohtuullisesti, mutta tarkempi vastaus vaatisi vielä lisätutkimusta. Tämä ehdottomasti motivoi adaptiivisuuden käyttöä myös jatkossa, sillä adaptiiviset tulokset vaikuttivat järkevämiltä, koska näiden painegradientin virhe pieneni odotetulla tavalla. Itseasiassa painegradientin erikoinen käyttäytyminen oli eräs pääsyy sille, miksi adaptiivisuutta lähdettiin kehittämään virtauslaskennan tarpeisiin. Samalla adaptiivisuus mahdollistaa niin tarkkoja tuloksia, että perinteisillä mallinnusmetodeilla vaaditaan kontrollitulavuuksien lukumäärällisesti jopa kertaluokkaa suurempia laskentaverkkoja, jolloin vaaditut resurssit ovat ainakin kertaluokkaa suurempia. Toisaalta adaptiivisuus vähentää paljon epävarmuutta numeeristen laskentatulosten analysoinnista, koska virheet ovat selkeästi näkyvillä, eikä niiden tulkinta välttämättä vaadi enää niin pitkää virtauslaskennan kokemusta. Tällöin virtauslaskija voi keskittyä vapaammin fysikaalisten reunaehtojen tai vaikkapa turbulenssimallien aiheuttaman mallivirheen pohtimiseen. Tutkimus osoittautui siis kaiken kaikkiaan erittäin hyödylliseksi.

LÄHTEET

Afif, M., Bergam, A., Mghazli, Z. & Verfürth, R. 2003, "A posteriori estimators for the finite volume discretization of an elliptic problem", *Numerical Algorithms*, vol. 34, no. 2, pp. 127-136.

Ainsworth, M. & Oden, J.T. 1993, "A unified approach to a posteriori error estimation using element residual methods", *Numerische Mathematik*, vol. 65, no. 1, pp. 23-50.

Ainsworth, M. & Oden, J.T. 1997, "A posteriori error estimation in finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 142, no. 1-2, pp. 1-88.

Carstensen, C., Lazarov, R. & Tomov, S. 2005, "Explicit and Averaging a Posteriori Error Estimates for Adaptive Finite Volume Methods", *SIAM Journal on Numerical Analysis*, vol. 42, no. 6, pp. 2496-2521.

Eltard-Larsen, B. 2016, *How to make a dynamicMotionRefineFvMesh class*, Chalmers University of Technology, Gothenburg. Available: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2015/BjarkeEltard-Larsen/dynamic-MotionRefineFvMesh_revised.pdf.

Ferziger, J.H. & Perić, M. 2002, *Computational Methods for Fluid Dynamics*, 3rd edn, Springer-Verlag, Berlin.

Foam-extend project, 2019, *Foam-extend Open Source CFD Toolbox*, the foam-extend project, web page. Available: <https://sourceforge.net/p/foam-extend/foam-extend-4.0/ci/master/tree/> [2019, 12.4.].

Greenshields, C.J. 2017, *OpenFOAM User Guide version 5.0*, The OpenFOAM Foundation Ltd, London.

Holzmann, T. 2018, *Mathematics, Numerics, Derivations and OpenFOAM(R)*, Holzmann CFD, Tussenhausen.

Holzmann, T. 2017, *Adaptive Mesh Refinement (Pseudo 2D)*, Holzmann CFD, web page. Available: <http://voluntary.holzmann-cfd.de/openfoam/openfoam-tutorials/dynamic-meshes/adaptive-mesh-refinement-pseudo-2d> [2019, 8.4.].

Jasak, H. 1996, *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*, Imperial College of Science, Technology and Medicine, Department of Mechanical Engineering, London. Available: https://spiral.imperial.ac.uk/bitstream/10044/1/8335/1/Hrvoje_Jasak-1996-PhD-Thesis.pdf.

Karlsson, J. 2012, *Implementing Anisotropic Adaptive Mesh Refinement in OpenFOAM*, Chalmers University of Technology, Department of Computer Science, Department of Computer Science, Gothenburg. Available: <http://publications.lib.chalmers.se/records/fulltext/174173/174173.pdf>.

Keough, S. 2014, *Optimising the Parallelisation of OpenFOAM Simulations*, Maritime Division DSTO Defence Science and Technology Organisation, Australia. Available: <https://apps.dtic.mil/docs/citations/ADA612337>.

Launder, B.E. & Sharma, B.I. 1974, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc", *Letters in Heat and Mass Transfer*, vol. 1, no. 2, pp. 131-137.

Lindblad, D. 2014, *Implementation and run-time mesh refinement for the k-omega SST DES turbulence model when applied to airfoils*, Chalmers University of Technology, Gothenburg. Available: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2013/DanielLindblad/k-Omega-SST-DES-Presentation.pdf.

Liu, F. 2017, *A Thorough Description of How Wall Functions Are Implemented In OpenFOAM*, Chalmers University of Technology, Gothenburg. Available: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016/FangqingLiu/openfoamFinal.pdf.

Menter, F.R. 1993, "Zonal Two Equation k-w Turbulence Models for Aerodynamic Flows", *24th AIAA Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Orlando, FL, United States, 6-9 Jul. 1993, pp. 1-21.

Menter, F.R., Kuntz, M. & Langtry, R. 2003, "Ten Years of Industrial Experience with the SST Turbulence Model", *Proceedings of the 4th International Symposium on Turbulence, Heat and Mass Transfer*, Begell House, New York, Wallingford, 12-17 Oct. 2003, pp. 625-632.

Mills, A.F. 2014, *Basic Heat and Mass Transfer*, 2nd edn, Pearson Education Limited, Harlow.

Moukalled, F., Mangani, L. & Darwish, M. 2016, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®*, Springer International Publishing AG, Cham.

Muzaferija, S. & Gosman, D. 1997, "Finite-Volume CFD Procedure and Adaptive Error Control Strategy for Grids of Arbitrary Topology", *Journal of Computational Physics*, vol. 138, no. 2, pp. 766-787.

Nieuwstadt, F.T.M., Boersma, B.J. & Westerweel, J. 2016, *Turbulence: Introduction to Theory and Applications of Turbulent Flows*, Springer International Publishing AG.

Oden, J.T., Wu, W. & Ainsworth, M. 1994, "An A Posteriori Error Estimate for Finite Element Approximations of the Navier–Stokes Equations", *Computer Methods in Applied Mechanics and Engineering*, vol. 111, no. 1-2, pp. 185-202.

OpenCFD Limited. 2018, *OpenFOAM The Open Source CFD Toolbox Programmer's Guide version v1812*, OpenCFD Limited, Bracknell.

OpenCFD Limited. 2009, *OpenFOAM The Open Source CFD Toolbox User Guide Version 1.6*, OpenCFD Limited, Bracknell.

Pope, S.B. 2000, *Turbulent Flows*, Cambridge University Press, Cambridge.

Roache, P.J. 1994, "Perspective: A Method For Uniform Reporting of Grid Refinement Studies", *Journal of Fluids Engineering*, vol. 116, no. 3, pp. 405-413.

Rumsey, C. 2015, *The Menter Shear Stress Transport Turbulence Model*, Langley Research Center, web page. Available: <https://turbmodels.larc.nasa.gov/sst.html#sst-std> [2018, 12.12.].

Sweby, P.K. 1984, "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws", *SIAM Journal on Numerical Analysis*, vol. 21, no. 5, pp. 995-1011.

Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A. & Tsamopoulos, J. 2017, "A critical analysis of some popular methods for the discretisation of the gradient operator in finite volume methods", *Physics of Fluids*, 127103, vol. 29, no. 12, pp. 1-40.

The OpenFOAM Foundation, 2019, *OpenFOAM v5.0 C++ Source Code Guide*. The OpenFOAM Foundation Ltd., web page. Available: <https://cpp.openfoam.org/v5/> [2019, 10.4.].

The OpenFOAM Foundation, 2016, *OpenFOAM 4.0 Released, Software and Development*. The OpenFOAM Foundation Ltd, web page. Available: <https://openfoam.org/release/4-0/> [2019, 12.4.].

Weller, H.G., Tabor, G., Jasak, H. & Fureby, C. 1998, "A tensorial approach to computational continuum mechanics using object-oriented techniques", *Computers in Physics*, vol. 12, no. 6, pp. 620-631.

Wilcox, D.C. 1993, *Turbulence modeling for CFD*, DCW Industries, Inc., La C nada.

Zienkiewicz, O.C., Taylor, R.L. & Zhu, J.Z. 2013, *The Finite Element Method: Its Basis and Fundamentals*, 7th edn, Butterworth-Heinemann, Oxford.