

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Martina Nobilo

ZAGREB, 2022

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

PERFORMANSE MULTIGRID ALGORITMA ZA LINEARNI SUSTAV
BRZINA-TLAK

Mentor:
prof. dr. sc. Hrvoje Jasak

Student:
Martina Nobile

ZAGREB, 2022

Iskrene zahvale idu profesoru Jasku za njegovo mentorstvo i dijeljenje svog znanja i stručnosti.

Posebno hvala Tessi Uroić koja je svojim znanjem, vodstvom i neizmjernom podrškom i najteže zadatke pretvorila u užitak učenja.

Hvala i doktorandima Luki, Mateju i Anji na pomoći i podršci - u takvoj atmosferi bi svatko poželio raditi svoj diplomski.

Posljednje i najvažnije, hvala mojim curama FSB-ovkama (Muri, Elizabeti, Martini i Niki) bez kojih bi zadnjih 6 godina bilo nemoguće preživjeti, hvala i svim prijateljima i bližnjima koji su uvijek bili tu uz mene (znate tko ste) i hvala mojim roditeljima i noni koji su bili, i još uvijek jesu, maksimalna podrška za sve moje ideje i ciljeve. Bez vas ovaj rad ne bi ugledao svjetlo dana.

Izjava

Izjavljujem da sam ovaj rad izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zagreb, prosinac 2022.

Martina Nobile



Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 22 - 6 / 1	
Ur.broj: 15 - 1703 - 22 -	

DIPLOMSKI ZADATAK

Student: **Martina Nobilo**

JMBAG: 0035208601

Naslov rada na hrvatskom jeziku: **Performanse multigrad algoritma za linearni sustav brzina-tlak**

Naslov rada na engleskom jeziku: **Performance of the multigrad algorithm for the velocity-pressure linear system**

Opis zadatka:

Struktura matrice linearnog sustava, dobivenog diskretizacijom lineariziranih parcijalnih diferencijalnih jednadžbi koje opisuju nestlačivo strujanje fluida, ovisi o odabranim shemama diskretizacije za pojedine članove jednadžbi. Pojedine sheme doprinose točnosti dobivenog rješenja, ali mogu izazvati nestabilnost konvergencije iterativnog algoritma za rješavanje linearnog sustava. Također, implicitnom spregom jednadžbi uvode se dodatna ograničenja kod odabira iterativnih algoritama s obzirom na spektralna svojstva matrice. U ovome radu, studentica će ispitati postavke multigrad algoritma za iterativno rješavanje linearnog sustava brzina-tlak i utjecaj na konvergenciju rješenja u paketu OpenFOAM.

U radu je potrebno:

- Proučiti literaturu i opisati spregu jednadžbi kontinuiteta i očuvanja količine gibanja te multigrad algoritam za rješavanje linearnog sustava;
- Prepoznati koje bi postavke multigrad algoritma mogle imati utjecaj na konvergenciju linearnog sustava;
- Odabrati geometriju za simulaciju strujanja fluida u kojemu dominira konvektivni prijenos veličina i izraditi proračunsku mrežu;
- Usporediti konvergenciju multigrad algoritma u odgovarajućoj nelinearnoj iteraciji za različite postavke algoritma;
- Ispitati utjecaj rubnih uvjeta jednadžbe tlaka i shema diskretizacije na konvergenciju multigrad algoritma;
- Usporediti konvergenciju multigrad algoritma na istoj geometriji za različite tipove proračunske mreže.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

Datum predaje rada:

Predviđeni datumi obrane:

29. rujna 2022.

1. prosinca 2022.

12. – 16. prosinca 2022.

Zadatak zadao:

Predsjednik Povjerenstva:

Prof. dr. sc. Hrvoje Jasak

Prof. dr. sc. Tanja Jurčević Lulić

Sadržaj

Sadržaj	v
Popis slika	viii
Popis tablica	x
Popis oznaka	xii
Sažetak	xiv
Abstract	xv
1. Uvod	1
1.1. Opseg rada	2
1.2. Struktura rada	3
2. Linearni sustav brzina-tlak	4
2.1. Uvod	4
2.2. Opis linearnog sustava jednadžbi	4
2.2.1. SIMPLE algoritam	6
2.3. Zaključak	8
3. Multigrid algoritam	10
3.1. Uvod	10
3.2. Veza matrice linearnog sustava i proračunske mreže	10

3.3. Algoritmi za rješavanje linearnog sustava	11
3.3.1. Metode stacionarne točke	11
3.3.2. Nekompletna LU faktorizacija	12
3.3.3. Motivacija nastanka multigrid algoritma	14
3.4. Multigrid algoritam	17
3.4.1. Struktura multigrid algoritma	17
3.4.2. Multigrid ciklus	19
3.5. Zaključak	24
4. Postavke multigrid algoritma	25
4.1. Uvod	25
4.2. Opis lokacije i promjena postavki multigrid algoritma	25
4.2.1. Odabir rješavača	26
4.2.2. Odabir multigrid ciklusa	26
4.2.3. Odabir načina formiranja matrice prolongacije i restrikcije	27
4.2.4. Odabir algoritma zaglađivača i njegove primjene	27
4.2.5. Odabir minimalnog broja jednadžbi na gruboj razini	28
4.2.6. Ograničavanje broja grubih razina	28
4.2.7. Ograničavanje broja iteracija	28
4.2.8. Postavke tolerancija	29
4.3. Promjena postavki unutar izvornog koda algoritma	29
4.3.1. Snaga veze između jednadžbi	29
4.3.2. Poredak jednadžbi na gruboj razini	30
4.4. Zaključak	30
5. Geometrija i proračunska mreža	32
5.1. Uvod	32
5.2. Geometrija	32
5.2.1. Proračunske mreže	33
5.2.2. Rubne plohe geometrije i rubni uvjeti	38
5.3. Zaključak	42
6. Rezultati	43
6.1. Uvod	43

6.2. Oznake	43
6.3. Poredak jednadžbi	45
6.3.1. <i>strongest first</i> poredak	45
6.3.2. <i>fine order</i> poredak	63
6.4. Snaga veza između jednadžbi	70
6.5. Promjena sume elemenata u retku matrice prolongacije	71
6.6. Broj primjene zaglađivača prije i nakon AMG rješavača	73
6.7. Zaključak	75
7. Diskusija	76
7.1. Uvod	76
7.2. Diskusija o rezultatima	76
7.2.1. <i>strongest first</i> poredak jednadžbi	76
7.2.2. <i>fine order</i> poredak jednadžbi	80
7.2.3. Snaga veza između jednadžbi	84
7.2.4. Promjena sume elemenata u retku matrice prolongacije	85
7.2.5. Broj primjene zaglađivača prije i nakon AMG rješavača	86
7.3. Sažetak rezultata	87
7.4. Zaključak	89
8. Zaključak	90
8.1. Zaključak	90
Literatura	93

Popis slika

3.1	Multigrid V-ciklus s dvije razine	19
3.2	Multigrid V-ciklus s više grubih razina	23
3.3	Multigrid W-ciklus s više grubih razina	24
3.4	Cijeli multigrid ciklus	24
4.1	Postavke multigrid algoritma unutar rječnika <i>fvSolution</i>	26
5.1	Odabrana geometrija s prikazanim točkama u $x-y$ ravnini	33
5.2	Proračunska mreža s 4800 kontrolnih volumena	34
5.3	Proračunska mreža s 19200 kontrolnih volumena	35
5.4	Dio proračunske mreže s 19200 kontrolnih volumena	36
5.5	Dio proračunske mreže (izlaz) s 25520 kontrolnih volumena	36
5.6	Proračunska mreža s 11046 kontrolnih volumena u obliku tetraedara . . .	37
5.7	Proračunska mreža s 6228 kontrolnih volumena u obliku tetraedara . . .	38
5.8	Oznake ploha/grupa ploha odabrane geometrije	39
6.1	Oznake položaja kontrolnih volumena i granice za zadavanje vrijednosti tlaka	44
6.2	Prikaz postavki multigrid algoritma unutar rječnika <i>fvSolution</i>	46
6.3	Konvergencija reziduala linearnog rješavača - mreža od 4800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 5	50
6.4	Konvergencija reziduala linearnog rješavača - mreža od 4800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 20	51

6.5	Konvergencija reziduala linearnog rješavača - mreža od 4800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 100	52
6.6	Konvergencija reziduala linearnog rješavača - mreža od 19200 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 5	53
6.7	Konvergencija reziduala linearnog rješavača - mreža od 19200 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 20	54
6.8	Konvergencija reziduala linearnog rješavača - mreža od 19200 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 100	55
6.9	Konvergencija reziduala linearnog rješavača - mreža od 76800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 5	56
6.10	Konvergencija reziduala linearnog rješavača - mreža od 76800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 20	57
6.11	Konvergencija reziduala linearnog rješavača - mreža od 76800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 100	58
6.12	Polje brzine na mreži od 76800 kontrolnih volumena u 24000. iteraciji	59
6.13	Polje tlaka na mreži od 76800 kontrolnih volumena u 24000. iteraciji	59
6.14	Konvergencija reziduala linearnog rješavača - mreža od 25520 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 20	63
6.15	Konvergencija reziduala linearnog rješavača s primjenom Gauss-Seidel i ILU algoritma na <i>fine order</i> poretku	65
6.16	Konvergencija reziduala linearnog rješavača s primjenom Gauss-Seidel algoritma na <i>fine order</i> poretku	66
6.17	Konvergencija reziduala linearnog rješavača s primjenom ILU algoritma na <i>fine order</i> poretku	66
6.18	Konvergencija reziduala linearnog rješavača s primjenom ILU algoritma na tetraedarskoj mreži	68
6.19	Konvergencija reziduala linearnog rješavača s primjenom ILU algoritma na tetraedarskoj mreži s većim kontrolnim volumenima na izlazu	69
6.20	Konvergencija reziduala linearnog rješavača s različitim brojem primjene ILU algoritma na <i>fine order</i> poretku	74
7.1	Preostale jednadžbe na gruboj razini sa snagom veze 0.1	84
7.2	Preostale jednadžbe na gruboj razini sa snagom veze 0.3	85

Popis tablica

5.1	Tablica rubnih uvjeta	41
6.1	Tablica rezultata simulacija s minimalnim brojem jednažbi na gruboj razini 5	47
6.2	Tablica rezultata simulacija s minimalnim brojem jednažbi na gruboj razini 20	48
6.3	Tablica rezultata simulacija s minimalnim brojem jednažbi na gruboj razini 100	49
6.4	Tablica usporedbe rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU i Gauss-Seidel algoritma s minimalnim brojem jednažbi na gruboj razini 20	60
6.5	Tablica rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU algoritma s minimalnim brojem jednažbi na gruboj razini 20	61
6.6	Tablica rezultata simulacija na mreži s 25520 kontrolnih volumena korištenjem Gauss-Seidel i ILU algoritma s minimalnim brojem jednažbi na gruboj razini 20	62
6.7	Tablica rezultata simulacija u 1000. iteraciji na mreži sa 76800 kontrolnih volumena korištenjem Gauss-Seidel i ILU algoritma s poretkom jednažbi kao u finoj matrici	64
6.8	Tablica rezultata simulacija u 10. iteraciji na tetraedarskim mrežama korištenjem ILU algoritma s poretkom jednažbi kao u finoj matrici	67

6.9	Tablica rezultata simulacija u 10. iteraciji na tetraedarskim mrežama korištenjem ILU algoritma s poretkom jednadžbi kao u finoj matrici	70
6.10	Tablica rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU i Gauss-Seidel algoritma s poretkom jednadžbi kao u finoj matrici i snagom veze 0.3	71
6.11	Tablica rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU algoritma s poretkom jednadžbi kao u finoj matrici i promjenom sume elemenata u retku matrice prolongacije	72
6.12	Usporedba rezultata simulacija prije i nakon promjene sume elemenata u retku matrice prolongacije	73
7.1	Tablica usporedbe kumulativne greške kontinuiteta na mreži od 76800 kontrolnih volumena s minimalnim brojem jednadžbi na gruboj razini 20	77
7.2	Tablica usporedbe broja elemenata u matrici	82
7.3	Tablica sažetka postavki algoritama zaglađivača	88
7.4	Tablica sažetka postavki multigrid algoritma	88

Popis oznaka

AAMG		Aggregation Algebraic Multigrid Algorithm	24
AMG		Algebraic Multigrid	10
ILU		Nepotpuna LU faktorizacija	2
RAS		Reynolds-Averaged Simulation	5
RDF		Računalna Dinamika Fluida	1
SAMG		Selection Algebraic Multigrid Algorithm	24
SIMPLE		Semi-Implicit Method for Pressure-Linked Equations	8
α_p	[-]	podrelaksacijski faktor za tlak	8
λ	[-]	svojstvena vrijednost	17
ν	[m ² /s]	kinematička viskoznost	5
A	[-]	matrica linearnog sustava	11
a_{ii}	[-]	dijagonalni element matrice linearnog sustava	8
a_{ij}	[-]	vandijagonalni element matrice linearnog sustava	8
$\tilde{\mathbf{A}}$	[-]	aproksimacija matrice linearnog sustava	14
D	[-]	dijagonala matrice linearnog sustava	12
d_N	[m]	udaljenost središta dva susjedna kontrolna volumena	8
D_u	[-]	skup dijagonalnih elemenata matrice linearnog sustava	8
e	[-]	vektor greške	17
ε	[m ² /s ³]	disipacija kinetičke energije turbulencije	5
H(u)	[-]	matrica koja sadrži samo vandijagonalni dio momentne matrice pomnožen s vrijednostima brzine	8
I	[-]	matrica identiteta	17

k	$[m^2/s^2]$	kinetička energija turbulencije	5
L	$[-]$	donjetrokutasta matrica	12
M	$[-]$	multigrad matrica	18
ω	$[1/s]$	specifična stopa disipacije kinetičke energije turbulencije . . .	5
P	$[-]$	matrica prolongacije	18
p	$[m^2/s^2]$	kinematički tlak	5
R	$[-]$	matrica restrikcije	18
r	$[-]$	vektor reziduala	17
\mathbf{r}_b	$[-]$	vektor desne strane linearnog sustava jednadžbi	8
S	$[-]$	smoother	24
\mathbf{s}_f	$[-]$	vektor normale na površinu	8
U	$[-]$	gornjetrokutasta matrica	12
u	$[m/s]$	vektor brzine	5
v	$[-]$	svojstveni vektor	17
w_{ij}	$[-]$	interpolacijski faktor matrice prolongacije	24
x	$[-]$	vektor rješenja linearnog sustava jednadžbi	12
x	$[-]$	os x koordinatnog sustava	33
y	$[-]$	os y koordinatnog sustava	33
z	$[-]$	os z koordinatnog sustava	33

Sažetak

S obzirom na sve veću upotrebu računalnih simulacija za rješavanje složenih problema u industriji i znanosti, potrebno je neprestano raditi na unaprjeđenju postojećih softvera kako bi matematičke jednadžbe koje opisuju stvarne fizikalne probleme bile što bolja reprezentacija stvarnosti. Zbog prirode matematičkih modela kojima se u računalnoj dinamici fluida (RDF) nastoje aproksimirati stvarni fizikalni problemi, prilikom pretvorbe diferencijalnih jednadžbi u diskretne jednadžbe i provođenja iterativnih postupaka javljaju se greške. U ovom radu proučavaju se performanse multigrid algoritma koji omogućava eliminiranje komponenti grešaka koje klasični iterativni algoritmi sporo eliminiraju. Izložena je teorijska pozadina klasičnih iterativnih algoritama i multigrid algoritma. Testirane su različite postavke multigrid algoritma na mrežama različitih gustoća i oblika kontrolnih volumena kako bi se donijeli zaključci o utjecaju veličine i oblika proračunske mreže na brzinu konvergencije multigrid algoritma. Također, testirana su dva različita algoritma u ulozi zaglađivača (algoritam koji zaglađuje rješenje prije i nakon multigrid ciklusa) - ILU i Gauss-Seidel - te su doneseni zaključci o njihovim performansama i uvjetima korištenja. Na kraju samog rada iznesena je sažeta tablica donesenih zaključaka kako bi se budućim korisnicima multigrid algoritma olakšalo njegovo korištenje.

Ključne riječi: RDF, selektivni algebarski multigrid, ILU, Gauss-Seidel

Abstract

Considering the increasing use of computer simulations to solve complex problems in industry and science, it is necessary to constantly work on the improvement of existing software so that the mathematical equations that describe real physical problems are a better representation of reality. Due to the nature of the mathematical models used in Computational Fluid Dynamics (CFD) to approximate real physical problems, errors occur when converting differential equations into discrete equations and performing iterative procedures. In this paper, the performance of the multigrid algorithm is studied, which enables the elimination of components of errors that classical iterative algorithms eliminate slowly. The theoretical background of classical iterative algorithms and the multigrid algorithm is presented. Different settings of the multigrid algorithm are tested on grids of different densities and shapes of control volumes in order to draw conclusions about the influence of the size and shape of the computational mesh on the speed of convergence of the multigrid algorithm. Also, two different algorithms were tested in the role of a smoother (an algorithm that smoothes the solution before and after the multigrid cycle) - ILU and Gauss-Seidel - and conclusions were drawn about their performance and conditions of use. At the end of the paper, a summary table of the conclusions was presented in order to make it easier for future users of the multigrid algorithm to use it.

Keywords: CFD, Selective Algebraic Multigrid, ILU, Gauss-Seidel

1 | Uvod

S obzirom na sve veću upotrebu računalnih simulacija za rješavanje složenih problema u industriji i znanosti, potrebno je neprestano raditi na unaprjeđenju postojećih softvera kako bi se simulacije što više približile stvarnosti, kako bi matematičke jednadžbe koje opisuju stvarne fizikalne probleme bile što bolja reprezentacija stvarnosti [1]. Ovaj rad fokusiran je na područje računalne dinamike fluida (RDF) i softverski paket pod nazivom OpenFOAM. OpenFOAM je slobodan softver otvorenog koda koji služi za rješavanje problema u mehanici kontinuuma koji se neprestano nadograđuje od strane istraživača i korisnika. Postoji još mnoštvo neistraženih postavki, algoritama i drugih dijelova softvera koji zahtijevaju testiranje i potvrdu performansi stoga se ovaj rad fokusira na jedno od dosad neistraženih područja.

Zbog prirode matematičkih modela kojima se u RDF nastoje aproksimirati stvarni fizikalni problemi, prilikom provođenja simulacija i dobivanja rješenja javljaju se greške. Te greške su posljedica iterativnih postupaka kojima se sustavi jednadžbi, nekoć diferencijalni, a sada pretvoreni u diskretne, rješavaju. U svrhu eliminacije nastalih pogrešaka koriste se određeni matematički algoritmi. U ovom radu proučavaju se performanse multigrid algoritma koji omogućava eliminiranje komponenti grešaka koje klasični iterativni algoritmi sporo eliminiraju. Diskretizacijom sustava jednadžbi koje se rješavaju na promatranoj proračunskoj mreži dobiva se matrica linearnog sustava koja opisuje odnose između kontrolnih volumena. Multigrid algoritam radi na principu pretvaranja fine matrice (matrice koja sadrži sve jednadžbe linearnog sustava) u grubu matricu (matricu koja sadrži samo odabrane jednadžbe) kako bi se svođenjem sustava na grubu nivo izračunala greška, točnije korekcija koja se potom koristi za korekciju rješenja jed-

nadžbi fine matrice. Izračunata greška koja je sada u odgovarajućem formatu može se eliminirati korištenjem posebnih algoritama.

U RDF učestalo je tretiranje nelinearnih i linearnih rješavača po principu *crne kutije*, bez pravog razumijevanja diskretnih jednadžbi, strukture linearnog sustava, a često i fizike problema iz koje korištene jednadžbe proizlaze. Cilj ovog rada je analizom i testiranjem multigrid algoritma doći do zaključaka o tome kako pojedine postavke algoritma utječu (ili ne utječu) na formiranje matrice linearnog sustava, konvergenciju rješenja te kako postavke rješavača utječu na performanse multigrid algoritma. Razumijevanje procesa kojim se dolazi od zadavanja uvjeta simulacije do konačnog rješenja omogućava kvalitetniju analizu točnosti dobivenih rješenja i skraćuje vrijeme potrebno za pronalaženje odgovarajućih postavki rješavača za zadani problem. Stoga je na kraju ovog rada iznesena detaljna analiza rezultata te sažeti prikaz postavki multigrid algoritma i utjecaja koji svaka od proučavanih postavki ima na linearni sustav i konačno rješenje kako bi se pružio uvid u *crnu kutiju* i olakšalo korištenje algoritma budućim korisnicima.

1.1. Opseg rada

U ovom radu testirat će se postojeće pretpostavke o radu Multigrid algoritma predstavljenom u [2] te istražiti dosad neispitani utjecaji različitih postavki algoritma. Testiranje će se provoditi na jednostavnoj dvodimenzionalnoj geometriji - cijev s proširenjem - na mrežama različite gustoće (različitog broja kontrolnih volumena) kako bi se vidio utjecaj koji veličina, oblik i raspored kontrolnih volumena imaju na rezultate. Provest će se stacionarne simulacije koristeći SimpleFOAM algoritam pokretan u foam-extend inačici OpenFOAM solvera. Pošto multigrid algoritam ne radi samostalno već u kombinaciji s drugim algoritmima, ispitat će se utjecaj postavki multigrid algoritma na rješenja dobivena korištenjem Gauss-Seidel i ILU (Incomplete Lower-Upper Factorization) algoritama kao takozvanih zaglađivača (*smoothers*) koji će biti detaljno objašnjeni u sljedećim poglavljima rada. Na kraju rada bit će dan sažeti pregled dobivenih rezultata i donesenih zaključaka u tabličnom prikazu.

1.2. Struktura rada

Ovaj rad sastoji se od teoretskog dijela u kojem će biti objašnjen linearni sustav brzina-tlak koji se primjenjuje na promatrani problem, opisat će se algoritmi za rješavanje linearnog sustava među kojima je i multigrid algoritam koji je tema ovog rada. Potom će biti navedene i objašnjene postavke multigrid algoritma te geometrija i mreže na kojima će se multigrid algoritam testirati. U drugom dijelu rada bit će prikazani dobiveni rezultati popraćeni diskusijom i sažetkom rezultata. Na kraju rada bit će donesen zaključak na temelju dobivenih rezultata.

2 | Linearni sustav brzina-tlak

2.1. Uvod

U prethodnom poglavlju ukratko je predstavljena tema rada i svrha Multigrid algoritma. Kako bi se upoznalo sa samim algoritmom potrebno je prije toga predstaviti sustav jednadžbi koji opisuje proučavani problem. U ovom poglavlju opisan je sustav spregnutih jednadžbi brzine i tlaka primijenjen na odabranoj geometriji te rješavanje takvog sustava.

2.2. Opis linearnog sustava jednadžbi

Strujanje za koje će se sustav postavljati je stacionarno, nestlačivo i jednofazno. Sustav jednadžbi sastoji se od jednadžbe očuvanja mase (jednadžbe kontinuiteta) (2.1) i jednadžbe očuvanja količine gibanja (momentne jednadžbe) (2.2):

$$\underbrace{\nabla \cdot \mathbf{u}}_{\text{divergencija brzine}} = 0 \quad (2.1)$$

$$\underbrace{\nabla \cdot (\mathbf{u}\mathbf{u}^T)}_{\text{konvekcijski član}} - \underbrace{\nabla \cdot (\nu \nabla \mathbf{u}^T)}_{\text{difuzijski član}} = - \underbrace{(\nabla p)^T}_{\text{gradijent tlaka}} \quad (2.2)$$

u kojima je \mathbf{u} brzina, p je kinematički tlak (statički tlak podijeljen s gustoćom) i ν je kinematička viskoznost. Sustav nosi naziv linearni jer se nelinearni dio jednadžbe očuvanja količine gibanja (konvekcijski član) eliminira linearizacijom. Turbulencija je

modelirana koristeći RAS model turbulencije (Reynolds-Averaged Simulation) u kojem se varijable zapisuju pomoću svojih glavnih i fluktuirajućih dijelova. Takav zapis varijabli dobiva se osrednjavanjem jednadžbi na malom vremenskom periodu. Sile nametnute turbulentnim fluktuacijama koje djeluju na glavni tok modelirane su korištenjem turbulentnih modela s dvije jednadžbe: $k - \varepsilon$ i $k - \omega$ modeli gdje je k kinetička energija turbulencije, ε je stopa disipacije kinetičke energije turbulencije i ω je specifična stopa disipacije [3]. U ovom radu korišten je $k - \varepsilon$ model.

Diskretizacija pojedinih članova jednadžbi 2.1 i 2.2 – navedena u [2] – provedena je prema sljedećim postavkama:

- za diskretizaciju divergensa konvekcijskog člana momentne jednadžbe koristi se *Gauss upwind* shema koja uzima vrijednosti iz kontrolnog volumena prije promatrane plohe, u smjeru suprotnom od smjera protoka kroz tu plohu, zbog čega se pokazala dobra posebice za početak simulacije gdje se vektori još usmjeravaju,
- za diskretizaciju difuzijskog člana momentne jednadžbe koristi se shema *Gauss linear* koja uzima vrijednosti iz susjednih kontrolnih volumena kojima promatrana ploha pripada i preslikava vrijednosti na plohu koristeći linearnu interpolaciju između vrijednosti u središtu kontrolnih volumena,
- za diskretizaciju gradijenta tlaka na desnoj strani momentne jednadžbe koristi se *Gauss linear* shema,
- za diskretizaciju Laplaceovog operatora u jednadžbi tlaka koristi se također *Gauss linear* shema.

U navedenom sustavu jednadžbi, 2.1 i 2.2, postoje dvije nepoznanice – brzina i tlak. Vidljivo je da se brzina pojavljuje kao nepoznanica u obje jednadžbe, dok se tlak pojavljuje u obliku gradijenta isključivo u momentnoj jednadžbi što znači da brzina ne ovisi o apsolutnoj vrijednosti tlaka, već o razlici tlaka u pojedinim točkama prostora. Pošto je za lakše rješavanje sustava brzina-tlak potrebno imati eksplicitnu jednadžbu za tlak, ona se mora izvesti iz jednadžbe kontinuiteta pri čemu se kontinuitet polja brzine osigurava nametanjem izračunatih vrijednosti tlaka – zadovoljavanje konzervativnosti polja brzine. Nakon formiranja sustava jednadžbi, odabire se način rješavanja istog. Sustav se može riješiti pristupanjem jednadžbama na dva različita načina. Jedan način je rješavanje

jednadžbe momenta i jednadžbe tlaka kao dva zasebna diskretna sustava, zbog čega nosi naziv odvojeni pristup, pri čemu se izračunate vrijednosti jedne varijable koriste za izračun druge. Drugi način je izvesti implicitno spregnuti sustav jednadžbi gdje se obje jednadžbe rješavaju istovremeno [4]. U ovom radu koristi se odvojeni pristup te je za rješavanje linearnog sustava brzina-tlak odabran SIMPLE algoritam.

2.2.1. SIMPLE algoritam

SIMPLE algoritam je jedna od metoda za rješavanje linearnog sustava brzina-tlak koju su osmislili Patankar i Spaldinga 1972. godine [5]. SIMPLE (eng. Semi-Implicit Method for Pressure-Linked Equations) označava djelomično implicitnu metodu za tlakom povezane jednadžbe kojom se iterativnim postupkom dolazi do rješenja stacionarnog linearnog sustava jednadžbi brzina-tlak [3]. Kako je već spomenuto, SIMPLE algoritam temelji se na odvojenom pristupu rješavanja sustava gdje se jednadžbe brzine i tlaka rješavaju odvojeno i sekvencijalno dok se ne dostigne postavljeni uvjet konvergencije rješenja. Jednadžba tlaka izvodi se izoliranjem brzine iz momentne jednadžbe te njezinom supstitucijom u jednadžbu kontinuiteta. Dobivena jednadžba tlaka glasi:

$$\nabla \cdot [(a_{ii}^u)^{-1} \nabla p] = \nabla \cdot [(a_{ii}^u)^{-1} \mathbf{H}(\mathbf{u})], \quad (2.3)$$

gdje je \mathbf{H} linearni operator, prema Jasak [6] koji se sastoji od vandijagonalnih članova momentne matrice \mathbf{A}_u , u kojoj su sadržani diskretizirani operatori konvekcije i difuzije, i vektora desne strane \mathbf{r}_b , u kojem su sadržani eksplicitni doprinosi postupka diskretizacije:

$$\mathbf{H}(\mathbf{u}) = \mathbf{r}_b - \sum_{j \neq i}^N a_{ij}^u \mathbf{u}_j. \quad (2.4)$$

Diskretizacija Laplaceovog operatora provodi se koristeći postupak suma-svih-površina za pretvorbu diferencijalnog operatora u diskretni i izgleda kako slijedi:

$$\int_{V_i} \nabla \cdot (D_u^{-1} \nabla p) dV = \sum_f (a_{ii}^{-1})_f s_f^T (\nabla p)_f, \quad (2.5)$$

gdje je difuzivnost jednaka inverzu dijagonalnih elemenata momentne matrice, Du^{-1} , a s_f^T je transponirani vektor površine. Kod diskretizacije Laplaceovog operatora u jednadžbi tlaka bitnu ulogu imaju zadani rubni uvjeti. U slučaju kada je zadana fiksna vrijednost tlaka na granici, gradijent tlaka na površini kontrolnog volumena računa se koristeći sljedeće vrijednosti:

$$(\nabla p)_f = \frac{p_b - p_i}{\|d_N\|}. \quad (2.6)$$

Fiksna vrijednost tlaka na granici u članu gradijenta tlaka doprinosi dijagonalnim elementima i vektoru desne strane \mathbf{r}_b . Kada je zadan gradijent tlaka na granici, ta se vrijednost gradijenta direktno koristi i ubacuje u vektor desne strane \mathbf{r}_b jednadžbe (2.4). Nakon formiranja jednadžbi SIMPLE algoritam sekvencijalno operira nad sustavom točno određenim redoslijedom operacija:

1. U svakoj nelinearnoj iteraciji prvo se rješava momentna jednadžba kako bi se dobio rješenje polja brzine pri čemu je potrebno eksplicitno izračunati gradijent tlaka iz pretpostavljenih vrijednosti (na početku iterativnog postupka) ili iz vrijednosti dobivenih u prethodnoj nelinearnoj iteraciji. Prilikom linearizacije konveksijskog člana javlja se volumetrički fluks ϕ koji se u ovom koraku uzima iz prethodne linearne iteracije. Također, sustav jednadžbi nakon diskretizacije sadrži i implicitnu podrelaksaciju potrebnu zbog nefizikalne komponente tlaka koja nastaje prilikom nametanja kontinuiteta polja brzine. Rješavanje momentne jednadžbe sa svim navedenim vrijednostima daje brzinu \mathbf{u}^* koja predstavlja privremeno rješenje i ne zadovoljava jednadžbu kontinuiteta.
2. Sljedeći korak je formiranje jednadžbe tlaka 2.3 izolirajući brzinu iz momentne jednadžbe i njezinom supstitucijom u jednadžbu kontinuiteta. Umjesto brzine \mathbf{u} uvrštava se njezino privremeno rješenje \mathbf{u}^* kako slijedi:

$$\nabla \cdot [(a_{ii}^u)^{-1} \nabla p] = \nabla \cdot [(a_{ii}^u)^{-1} H(\mathbf{u}^*)] \quad (2.7)$$

Rješenje dobivene jednadžbe tlaka je tlak p^* koji sadrži fizikalno polje tlaka koje dolazi od polja protoka i nefizikalno polje koje je odgovorno za nametanje kontinuiteta polja brzine i kompenzaciju greške u početnom polju tlaka. Dobiveno

polje tlaka se stoga eksplicitno podrelaksira kako bi se uklonila nefizikalna komponenta rješenja. Podrelaksacija se provodi tako da se nova vrijednost polja tlaka izračunava s vrlo malim udjelom polja tlaka dobivenog u ovom koraku dok se ostatak vrijednosti polja tlaka preuzima iz prethodne iteracije. Podrelaksacija tlaka navedena je u sljedećoj jednadžbi :

$$p^{(k)} = (1 - \alpha_p)p^{(k-1)} + \alpha_p p^* \quad (2.8)$$

gdje $p^{(k-1)}$ označava vrijednost polja tlaka u prethodnom vremenskom koraku, p^* vrijednost tlaka dobivenu iz jednadžbe tlaka i $0 < \alpha_p \leq 1$ podrelaksacijski faktor za tlak.

3. U posljednjem koraku polje brzine \mathbf{u}^* , izračunato u koraku 1 iz momentne jednadžbe, ispravlja se koristeći podrelaksirane vrijednosti polja tlaka iz koraka 2.

Sekvenca navedena 3 koraka ponavlja se sve dok linearni i nelinearni dio rješenja ne konvergira do zadanih tolerancija. Ponavljanje sekvence ovih koraka naziva se nelinearna iteracija dok se rješavanje pojedinačnih jednadžbi sustava naziva linearna iteracija. SIMPLE algoritam karakterizira još i štednja memorije pošto se nakon rješenja momentne jednadžbe isti prostor za pohranjivanje podataka koristi za pohranjivanje rješenja jednadžbe tlaka zato što su obje matrice rijetke i imaju isti položaj elemenata (eng. *sparsity pattern*).

2.3. Zaključak

U ovom poglavlju dan je pregled linearnog sustava jednadžbi koji opisuje proučavani problem kao i opis shema diskretizacije primijenjenih na pojedine članove navedenih jednadžbi u svrhu razumijevanja linearnog sustava na kojem se multigrid algoritam primjenjuje. Nadalje, detaljno je opisan SIMPLE algoritam čija je uloga rješavanje linearnog sustava jednadžbi formiranjem jednadžbe tlaka i sekvencijalnim ponavljanjem koraka algoritma.

Sljedeće poglavlje bavi se motivacijom i svrhom razvoja multigrid algoritma te sadrži detaljan redoslijed koraka koje algoritam provodi nad matricom linearnog sustava. Kako

bi se objasnila motivacija nastanka multigrid algoritma, bit će predstavljene i algoritmi stacionarne točke, njihove prednosti i mane u odnosu na multigrid te uloga u njegovom nadopunjavanju.

3 | Multigrid algoritam

3.1. Uvod

U ovom poglavlju bit će opisan algebarski multigrid linearni rješavač (eng. Algebraic Multigrid, AMG). Naziv algebarski dolazi od načina na koji multigrid algoritam radi točnije, AMG radnje izvršava isključivo na matrici \mathbf{A} linearnog sustava, dok algoritam nema nikakvo djelovanje na proračunsku mrežu, za razliku od geometrijskog multigrid algoritma. U svrhu pojašnjenja svrhe multigrid algoritma, prvo će biti objašnjena struktura matrice linearnog sustava \mathbf{A} , a potom metode rješavanja linearnog sustava pod nazivom metode stacionarne točke iz čijih ograničenja i nedostataka potom proizlazi potreba za multigrid linearnim rješavačem.

3.2. Veza matrice linearnog sustava i proračunske mreže

Svaka proračunska mreža u OpenFOAM-u sastoji se od određenog broja trodimenzionalnih kontrolnih volumena koji su preko svojih ploha povezani sa susjednim kontrolnim volumenima. Za svaki od kontrolnih volumena provodi se diskretizacija sustava linearnih jednadžbi postavljenih za proračunsku mrežu, i svaki kontrolni volumen je u matrici linearnog sustava \mathbf{A} predstavljen jednim retkom. Indeks retka i predstavlja indeks kontrolnog volumena. Dijagonalni element u retku, $a_{i,i}$, predstavlja utjecaj kontrolnog volumena „samog na sebe“ dok vandijagonalni elementi, $a_{i,j}$, predstavljaju utjecaj susjednih kontrolnih volumena. Stupac vandijagonalnog elementa jednak je indeksu su-

sjednog kontrolnog volumena. Veličina vandijagonalnog elementa $|a_{i,j}|$ pokazuje utjecaj susjednog kontrolnog volumena, odnosno snagu veze između dviju jednadžbi. Matrica linearnog sustava \mathbf{A} izgleda kako slijedi:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{2,1} & a_{2,2} & a_{2,3} & & & & & \\ a_{3,1} & a_{3,2} & a_{3,3} & & & & & \\ \cdot & & & \cdot & & & & \\ \cdot & & & & \cdot & & & \\ \cdot & & & & & \cdot & & \\ \cdot & & & & & & \cdot & \\ a_{i,j} & & & & & & & a_{i,i} \end{bmatrix} \quad (3.1)$$

3.3. Algoritmi za rješavanje linearnog sustava

3.3.1. Metode stacionarne točke

Algoritmi za rješavanje linearnog sustava jednadžbi su iterativne metode rješavanja linearnog sustava $\mathbf{Ax} = \mathbf{b}$. Postoji širok spektar takvih metoda od kojih će samo neke, bitne za ovaj rad, biti spomenute u ovom radu. Jedna skupina metoda su metode stacionarne točke. Primjeri takvih metoda su Jacobi i Gauss-Seidel algoritmi [7] od kojih će posljednji biti ukratko objašnjen. Gauss-Seidel radi na principu razdvajanja matrice linearnog sustava \mathbf{A} na sumu dijagonalnog (\mathbf{D}) i vandijagonalnog dijela. Neditijagonalni dio se potom dijeli na gornjetrokutastu (\mathbf{U}) i donjetrokutastu (\mathbf{L}) matricu:

$$\mathbf{A} = \mathbf{D} + \mathbf{U} + \mathbf{L}. \quad (3.2)$$

Kada se linearni sustav napiše pomoću dijelova matrice i primjeni se svojstvo distributivnosti prilikom množenja matrice i vektora, dobiju se sljedeći izrazi:

$$(\mathbf{D} + \mathbf{U} + \mathbf{L})\mathbf{x} = \mathbf{b} \quad (3.3)$$

$$\mathbf{D}\mathbf{x} + \mathbf{U}\mathbf{x} + \mathbf{L}\mathbf{x} = \mathbf{b}. \quad (3.4)$$

Postoje dva načina kojima Gauss-Seidel prolazi kroz matricu. Jedan način je propagacija algoritma od prve do posljednje komponente rješenja (odozgo prema dolje), takozvana *forward technique*. Komponente rješenja, x_i , koje množe gornjetrokutastu matricu eksplicitno se tretiraju koristeći prethodno izračunate vrijednosti:

$$\mathbf{x}^{novi} = \mathbf{D}^{-1}(\mathbf{b}_i - \mathbf{L}\mathbf{x}^{novi} - \mathbf{U}\mathbf{x}^{stari}), \quad (3.5)$$

gdje potencije *novi* i *stari* označavaju vrijednosti u trenutnoj i prošloj iteraciji, a indeks i označava komponente vektora rješenja. Drugi način je propagacija algoritma od posljednje do prve komponente rješenja (odozdo prema gore), *backward technique*, koji funkcionira isto kao i *forward technique* osim što komponente rješenja sada množe donjetrokutastu matricu. Jacobi metoda funkcionira na isti način kao Gauss-Seidel osim što ne koristi nove dobivene vrijednosti komponenti rješenja, \mathbf{x}_{novi} , već stare vrijednosti zbog čega sporije konvergira od Gauss-Seidel metode. U trenutku kada algoritam nađe točno rješenje ($\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n$), te komponente se više ne rješavaju – one postaju *stacionarne točke*, otkud dolazi i naziv ove skupine metoda.

3.3.2. Nekompletna LU faktorizacija

Sljedeća metoda rješavanje linearnog sustava bitna za ovaj rad jest nekompletna LU faktorizacija (eng. ILU - Incomplete Lower-Upper factorization). Metoda uzima matricu i rastavlja ju na produkt gornje i donjetrokutaste matrice korištenjem Gaussovih eliminacija. Gaussova eliminacija je direktna metoda rješavanja linearnog sustava koja eliminira elemente donjetrokutaste matrice kako bi se dobila gornjetrokutasta matrica nakon čega je izračun rješenja sustava trivijalan [7]. Slijed koraka Gaussove eliminacije izgleda kako slijedi:

- Za svaki novi red matrice, uđi u sve prethodne redove i izračunaj multiplikatore (omjer svakog vandijagonalnog i dijagonalnog člana u tom retku).
- Prođi kroz sve elemente retka matrice u kojem se nalaziš i ažuriraj njegove elemente s dobivenim multiplikatorima (doprinosima prethodnih redaka).

Dobiveni multiplikatori za eliminaciju elemenata donjetrokutaste matrice se mogu postaviti na odgovarajuća mjesta u donjetrokutastu matricu koja ima jedinice kao elemente dijagonale. Matrica \mathbf{A} se tada može zapisati kao produkt dobivene donje i gornjetrokutaste matrice:

$$\mathbf{A} = \mathbf{L}_A \mathbf{U}_A. \quad (3.6)$$

LU faktorizacija rijetke matrice će općenito dati guste faktore, odnosno povećat će broj elemenata matrice koji su različiti od 0, što nije poželjno u smislu reduciranja potrebe za memorijom. Za prekondicioniranje matrice (pripremu za njezino rješavanje) je dovoljno koristiti aproksimaciju matrice \mathbf{A} , $\tilde{\mathbf{A}}$, gdje nije potrebno računati točne multiplikatore. Produkt donje i gornjetrokutaste matrice tada nije jednak originalnoj matrici \mathbf{A} i postoji matrica greške \mathbf{R} :

$$\mathbf{A} = \underbrace{\tilde{\mathbf{A}}}_{\mathbf{LU}} \mathbf{R}, \quad (3.7)$$

gdje su \mathbf{L} i \mathbf{U} aproksimacije donje i gornjetrokutaste matrice. Nepotpuna LU faktorizacija sastoji se od modificiranih Gaussovih eliminacija gdje se neki od vandijagonalnih elemenata gornje i donjetrokutaste matrice ispuštaju, a ovisno o načinu odabira koji će se elementi ispustiti postoje dva oblika ILU algoritma - dinamički ILU gdje se koristi granična vrijednost i statički ILU gdje se koristi predodređeni uzorak popunjavanja matrice. Dinamički ILU algoritam ispušta elemente iz matrice ili ih zadržava ovisno o tome jesu li dovoljno veliki s obzirom na graničnu vrijednost, ali se teško kontrolira popunjenost matrice. Statički ILU algoritam za popunjavanje matrice koristi isti uzorak popunjenosti matrice \mathbf{A} koji se može i proširivati stoga je prikladan za kontrolu količine podataka koji se pohranjuju. U sklopu ovog rada koristit će se statički ILU algoritam. Postoje dvije varijante ILU algoritma ovisno o načinu rješavanja gornje i donjetrokutaste matrice:

- IKJ varijanta - retci faktora \mathbf{L} i \mathbf{U} se sekvencijalno generiraju koristeći dijagonalne elemente svih prethodnih redaka za eliminaciju vandijagonalnih elemenata promatranog retka donjetrokutaste matrice,
- KIJ varijanta - modificira više redova koristeći elemente iz jednog retka.

IKJ varijanta može biti proračunski veoma skupa stoga je razvijena praktičnija verzija pod nazivom Crout LU faktorizacija koja u k -tom koraku izračunava k -ti redak u \mathbf{U} i k -ti stupac u \mathbf{L} simultano. Statička Crout ILU faktorizacija se koristi u ovom radu gdje je uzorak popunjenosti \mathbf{L} i \mathbf{U} jednak kao u matrici \mathbf{A} što se naziva razinom popunjenosti 0 (ILUC0). Upotreba ILUC0 znači da su popunjena (ne-nula) samo mjesta s indeksom stupca koji pripada prvim susjedima kontrolnog volumena. Razina popunjenosti 1 i više razine proširuju popunjenost matrice i na mjesta s indeksom stupca koji pripada susjedima prvih susjeda kontrolnog volumena, itd.

Rezultat nekompletne ILU faktorizacija ovisi o poretku jednadžbi u matrici gdje će \mathbf{L} i \mathbf{U} biti vrpčaste matrice ako je i \mathbf{A} vrpčasta matrica [8]. Dakle, nepotpuna Crout ILU faktorizacija se radi prema predodređenom uzorku popunjenosti matrice. Nakon završetka faktorizacije, rješavaju se dva linearna sustava jednadžbi - jedan za gornjetrokutastu i drugi za donjetrokutastu matricu. Prvo se rješava supstitucija unaprijed koristeći donjetrokutastu matricu

$$\mathbf{L}\mathbf{y} = \mathbf{b}, \quad (3.8)$$

a potom supstitucija unatrag koristeći gornjetrokutastu matricu

$$\mathbf{U}\mathbf{x} = \mathbf{y}. \quad (3.9)$$

.

3.3.3. Motivacija nastanka multigrid algoritma

Za prekid iterativnog postupka, odnosno kao kriterij konvergencije aproksimacije rješenja koristi se rezidual:

$$\mathbf{r}^{(k)} = \mathbf{A}\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} \quad (3.10)$$

gdje je \mathbf{r} vektor reziduala, \mathbf{b} je vektor desne strane, a \mathbf{x} je aproksimacija rješenja trenutne nelinearne iteracije (koju označava indeks k). Matrica \mathbf{A} i vektor desne strane \mathbf{b} formiraju se na početku svake nelinearne iteracije i linearne iteracije, u kojima se izračunavaju aproksimacije rješenja sustava \mathbf{x} , nemaju utjecaj na njih. Rezidual stoga govori o tome koliko je naše trenutno rješenje daleko od stvarnog rješenja sustava. Greška:

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)} \quad (3.11)$$

gdje je \mathbf{e} vektor greške, a \mathbf{x} točno rješenje sustava, je nepoznata zbog čega se ona ne može koristiti kao kriterij konvergencije rješenja. Veza između vektora reziduala i vektora greške glasi:

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}^{(k)} = \mathbf{A}(\mathbf{x} - \mathbf{x}^{(k)}) = \mathbf{A}\mathbf{e}^{(k)} \quad (3.12)$$

što znači da je rezidual zapravo greška transformirana u prostor matrice \mathbf{A} .

Međutim, mali iznos reziduala ne znači nužno da je i greška mala. Kako bi se navedena tvrdnja objasnila, potrebno je proučiti vektore greške i reziduala u istom vektorskom prostoru – prostoru matrice \mathbf{A} . Rezidual se već nalazi u prostoru matrice \mathbf{A} , dok se transformacija vektora greške provodi pomoću svojstvenih vektora matrice \mathbf{A} . Svojstveni vektori matrice \mathbf{A} , ako je ona simetrično pozitivno-definitna, su vektori koje matrica ne može zarotirati točnije, može ih produljiti ili skratiti, ali ne može mijenjati kutove između njih. Nadalje, svojstveni vektori su linearno nezavisni (njihov skalarni produkt jednak je 0). Ako je matrica \mathbf{A} simetrično pozitivno-definitna, onda postoji n linearno nezavisnih svojstvenih vektora \mathbf{v} za koje vrijedi:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad (3.13)$$

gdje je λ skalar i svojstvena vrijednost matrice \mathbf{A} . Promatrajući trodimenzionalni prostor, vektor greške \mathbf{e} može se projicirati na svojstveni vektor matrice \mathbf{A} prema sljedećem izrazu:

$$\mathbf{e} = a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3 \quad (3.14)$$

i potom transformirati u prostor matrice \mathbf{A} množenjem s matricom \mathbf{A} :

$$\mathbf{r} = \mathbf{A}(a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3) = a\mathbf{A}\mathbf{v}_1 + b\mathbf{A}\mathbf{v}_2 + c\mathbf{A}\mathbf{v}_3 = a\lambda_1\mathbf{v}_1 + b\lambda_2\mathbf{v}_2 + c\lambda_3\mathbf{v}_3. \quad (3.15)$$

Iz gornje jednadžbe vidljivo je da je greška u prostoru matrice \mathbf{A} skalirana sa svojstvenim vrijednostima njezinih svojstvenih vektora. Kada su $\lambda > 1$, svojstveni vektor \mathbf{v} se

izdužuje i rezidual je velik, dok greška može biti mala. Kada su $\lambda < 1$, svojstveni vektor \mathbf{v} se skraćuje i rezidual je malen, dok greška može biti velika. Rezidual će dobro ilustrirati udaljenost od točnog rješenja kada su $\lambda \approx 1$ ili je matrica \mathbf{A} matrica identiteta (sadrži jedinice na dijagonali i nule na mjestima svih vandijagonalnih članova).

Vraćajući se na Gauss-Seidel metodu čija se iteracija može zapisati u obliku:

$$\mathbf{x}^{novi} = (\mathbf{I} - \mathbf{A})\mathbf{x}^{stari} + \mathbf{b}, \quad (3.16)$$

može se izvesti ovisnost greške u novoj iteraciji o vrijednosti greške u prošloj iteraciji:

$$\mathbf{e}^{nova} = (\mathbf{I} - \mathbf{A})\mathbf{e}^{stara} = \mathbf{e}^{stara} - \mathbf{A}\mathbf{e}^{stara}, \quad (3.17)$$

koja prikazana pomoću svojstvenih vektora glasi:

$$\begin{aligned} \mathbf{e}^{nova} &= a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3 - \mathbf{A}(a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3) \\ &= a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3 - (a\lambda_1\mathbf{v}_1 + b\lambda_2\mathbf{v}_2 + c\lambda_3\mathbf{v}_3) \end{aligned} \quad (3.18)$$

iz čega je vidljivo da, kada su svojstvene vrijednosti $\lambda \approx 1$, \mathbf{e}^{nova} teži nuli, odnosno konvergencija će biti brza. U slučaju svojstvenih vrijednosti $\lambda \approx 0$, \mathbf{e}^{nova} se neće puno smanjiti u odnosu na \mathbf{e}^{stara} , odnosno konvergencija će biti veoma spora. Zaključno, komponente greške \mathbf{e} projicirane na svojstveni vektor \mathbf{v} čije su svojstvene vrijednosti $\lambda \approx 1$ brzo se eliminiraju i nazivaju se visokofrekventne greške. Visokofrekventno naglašava činjenicu da takve greške koreliraju između bliskih članova u matrici stoga ih algoritmi stacionarne točke mogu ukloniti. Komponente greške \mathbf{e} projicirane na svojstveni vektor \mathbf{v} čije su svojstvene vrijednosti $\lambda \approx 0$ sporo se eliminiraju (ili se uopće ne eliminiraju) i nazivaju se niskofrekventne greške. Niskofrekventno označava da takve greške ne koreliraju „blizu“ u matrici i njih algoritmi stacionarne točke vrlo sporo eliminiraju. Iz navedenog problema proizlazi motivacija za multigrid algoritam koji visokofrekventne komponente greške prepoznaje i eliminira, a niskofrekventne greške pretvara u visokofrekventne koje potom mogu biti eliminirane algoritmima stacionarne točke.

Nadalje, pošto su matrice koje proizađu iz diskretizacije konačnih volumena rijetke (većina elemenata je 0), pojedina komponenta rješenja ovisi samo o nekoliko drugih komponenti rješenja. Prouči li se navedena tvrdnja sa strane proračunske mreže, na vrijednost u pojedinom kontrolnom volumenu utjecat će samo vrijednosti iz kontrolnih volumena u najbližem susjedstvu. Metode stacionarne točke, kao Gauss-Seidel,

koriste informacije u blizini elementa čije rješenje računaju zbog čega će konvergencija rješenja biti brza samo ako je matrica dijagonalno dominantna (ako je suma vandijagonalnih elemenata u svakom retku manja od veličine dijagonalnog elementa), dok će se brzina konvergencije smanjivati za matrice koje nisu snažno dijagonalno dominantne jer je tada utjecaj elemenata u susjedstvu jači i širenje informacija je otežano. Multigrid algoritam omogućava globalno širenje informacija stvaranjem hijerarhije grubih matrica (matrice s manje redaka i stupaca) na način da jednadžbe koje u finoj matrici nisu imale komunikaciju (vandijagonalni elementi su 0), u gruboj matrici mogu postati susjedi (vandijagonalni članovi različiti od 0) zbog čega se utjecaj njihovih rješenja širi.

3.4. Multigrid algoritam

3.4.1. Struktura multigrid algoritma

U objašnjenju strukture multigrid algoritma zbog jednostavnosti koristit će se algoritam sa samo dvije razine (jednom finom i jednom grubom), dok se u stvarnosti korištenjem multigrid algoritma formira više grubih razina [9]. Struktura algoritma, prema [2] izgleda kako slijedi:

1. Izračunavanje aproksimacije rješenja na finoj razini:

$$\mathbf{A}_{n \times n}^F \bar{\mathbf{x}}^F = \mathbf{b}, \quad (3.19)$$

gdje je $\mathbf{A}_{n \times n}^F$ matrica na finoj razini, $n \times n$ govori o dimenzijama matrice (broju jednadžbi koje se rješavaju), $\bar{\mathbf{x}}^F$ je aproksimacija rješenja na finoj razini i \mathbf{b} vektor desne strane.

2. Korištenjem dobivene aproksimacije rješenja iz koraka 1, izračunava se rezidual na finoj razini:

$$\mathbf{r}^F = \mathbf{b} - \mathbf{A}_{n \times n}^F \bar{\mathbf{x}}^F. \quad (3.20)$$

3. Kreiranje grube matrice $\mathbf{A}_{m \times m}^C$ temeljem određenih kriterija gdje je m broj jednadžbi na gruboj razini manji od n broja jednadžbi na finoj razini ($m < n$). Postupak odabira jednadžbi koja se rješavaju na gruboj (eng. *coarse*) razini naziva se *coarsening*.
4. Transformiranje reziduala s fine razine \mathbf{r}^F na grubu razinu \mathbf{r}^C pomoću matrice restrikcije $\mathbf{R}_{m \times n}$:

$$\mathbf{r}^C = \mathbf{R}_{m \times n} \mathbf{r}^F. \quad (3.21)$$

5. Rješavanje jednadžbe korekcije na gruboj razini:

$$\mathbf{A}_{m \times m}^C \mathbf{e}^C = \mathbf{r}^C, \quad (3.22)$$

gdje je \mathbf{e}^C pogreška na gruboj razini koja se zbog svoje uloge u sljedećem koraku algoritma još naziva i korekcija.

6. Transformacija greške s grube razine \mathbf{e}^C na finu razinu \mathbf{e}^F pomoću matrice prolongacije $\mathbf{P}_{n \times m}$:

$$\mathbf{e}^F = \mathbf{P}_{n \times m} \mathbf{e}^C. \quad (3.23)$$

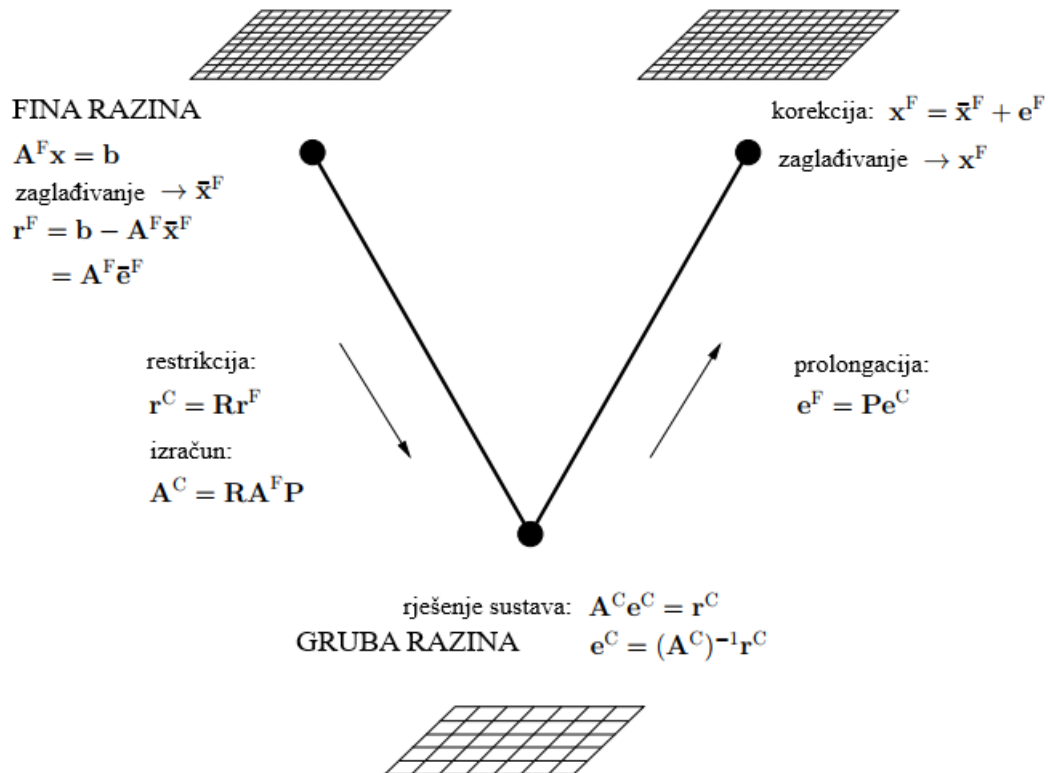
7. Ispravljanje rješenja dobivenog na finoj razini, \mathbf{x}^F , koristeći prolongiranu grešku \mathbf{e}^F :

$$\bar{\mathbf{x}}^F = \bar{\mathbf{x}}^F + \mathbf{e}^F. \quad (3.24)$$

U narednom poglavlju bit će dan detaljniji opis V-ciklusa s jednom grubom razinom i objašnjenje multigrid matrice \mathbf{M} .

3.4.2. Multigrid ciklus

V-ciklus s dvije razine, finom i grubom, i naznačenim algebarskim operacijama prikazan je na sljedećoj slici:



Slika 3.1: Multigrid V-ciklus s dvije razine

Objašnjenje postupka prikazanog na slici 3.1 je sljedeće:

- Fina razina
 - Izračun aproksimacije rješenja $\bar{\mathbf{x}}^F$ sustava $\mathbf{A}^F \mathbf{x} = \mathbf{b}$ koristeći metodu stacionarne točke.
 - Izračun reziduala na temelju dobivene aproksimacije rješenja:
 $\mathbf{r}^F = \mathbf{b} - \mathbf{A}^F \bar{\mathbf{x}}^F = \mathbf{A}^F \bar{\mathbf{e}}^F$.
 - Restrikcija reziduala (smanjivanje broja komponenti vektora reziduala):
 $\mathbf{r}^C = \mathbf{R} \mathbf{r}^F$.

- Gruba razina
 - Rješavanje korekcijske jednadžbe $\mathbf{A}^C \mathbf{e}^C = \mathbf{r}^C$ kako bi se dobila greška \mathbf{e}^C na gruboj razini.
 - Prolongacija greške (korekcije) (povećavanje broja komponenti vektora greške interpolacijom rješenja jednadžbi s grube razine): $\mathbf{e}^F = \mathbf{P}\mathbf{e}^C$.
- Fina razina
 - Ispravak rješenja : $\mathbf{x}^F = \bar{\mathbf{x}}^F - \mathbf{e}^F$.
 - Zaglađivanje rješenja (eng. *smoothing*) koristeći metodu stacionarne točke.

Kada se promatraju sve operacije provedene na greški \mathbf{e}^F kroz čitav V-ciklus, može se izvesti sljedeći izraz [8]:

$$\underbrace{\mathbf{P}(\mathbf{A}^C)^{(-1)}\mathbf{R}}_{\text{rješenje grubog sustava}} \underbrace{\underbrace{\mathbf{A}^F \bar{\mathbf{e}}^F}_{\text{rezidual fina razina}}}_{\text{restrikcija reziduala}} = \mathbf{e}^F. \quad (3.25)$$

prolongacija korekcije

Iz izraza (3.25) može se uočiti multigrid matrica \mathbf{M} :

$$\mathbf{M} := \mathbf{P}(\mathbf{A}^C)^{(-1)}\mathbf{R}\mathbf{A}^F. \quad (3.26)$$

koja se sastoji od matrice linearnog sustava na finoj, \mathbf{A}^F , i gruboj razini, \mathbf{A}^C , te matrice prolongacije \mathbf{P} i restrikcije \mathbf{R} . Matrica linearnog sustava \mathbf{A} objašnjena je u prethodnim dijelovima rada, ali matrice prolongacije i restrikcije zahtijevaju detaljnije pojašnjenje. Matrica prolongacije \mathbf{P} služi za prebacivanje greške, koja je izračunata na gruboj razini, na finu razinu. Greška na gruboj razini ima manje dimenzije (sadrži manje komponenti), a za njezino korektivno djelovanje na rješenje potrebno je uskladiti broj komponenti vektora greške (korekcije) i vektora rješenja. Matrica prolongacije \mathbf{P} ima onoliko redova koliki je broj jednadžbi u finoj matrici i onoliko stupaca koliki je broj jednadžbi u gruboj matrici. Elementi matrice prolongacije su interpolacijski faktori. Ideja je da će svaka jednadžba na finoj razini dobiti korekciju s grube razine od svojih grubih

utjecaja (susjeda) preko težinske interpolacije. Matrica prolongacije množi vektor greške s grube razine i ovisno o interpolacijskim faktorima provodi linearnu interpolaciju između postojećih komponenti greške kako bi se dobio odgovarajući broj komponenti vektora greške na finoj razini. Primjer matrice prolongacije vidljiv je na sljedećoj slici:

$$\mathbf{P} = \begin{bmatrix} w_{i,j} & w_{i,j} & w_{i,j} \\ 1 & 0 & 0 \\ w_{i,j} & 0 & w_{i,j} \\ 0 & w_{i,j} & w_{i,j} \\ w_{i,j} & w_{i,j} & 0 \end{bmatrix} \quad (3.27)$$

gdje indeks i označava broj retka matrice (jednadžu na finoj razini), a indeks j broj stupca matrice (jednadžbu na gruboj razini koja je utjecaj promatrane fine jednadžbe). Na mjestima na kojima se nalazi 1, jednadžba na finoj razini uzima korekciju sama od sebe pošto se ona ujedno nalazi i na grubom nivou (odabrana je u procesu *coarsening-a*). Na mjestima gdje su 0, jednadžba na finoj razini ne uzima korekciju od jednadžbe na gruboj razini jer ona nije njezin utjecaj. Ako je na finoj razini matrica dijagonalno dominantna (veličina elementa na dijagonali je veća od sume veličina vandijagonalnih elemenata), onda je $\sum_i w_{i,j} > 1$ u odgovarajućem retku matrice prolongacije \mathbf{P} .

Matrica restrikcije \mathbf{R} , naprotiv, služi prilagođavanju dimenzija vektora reziduala, dobivenog na finoj razini, dimenzijama grube matrice. Matrica restrikcije \mathbf{R} je transponirana matrica prolongacije \mathbf{P} , $\mathbf{R} = \mathbf{P}^T$, stoga neće biti posebno objašnjena. Uobičajeno je da se prvo formira matrica prolongacije \mathbf{P} , a potom se matrica restrikcije \mathbf{R} dobiva transponiranjem matrice \mathbf{P} . Matrice prolongacije i restrikcije dobivene metodom linearne interpolacije karakteriziraju selektivni algebarski multigrid algoritam (eng. Selection Algebraic Multigrid Algorithm, SAMG)[10]. Agregacijski algebarski multigrid algoritam (eng. Aggregation Algebraic Multigrid Algorithm, AAMG) izgrađuje matrice prolongacije i restrikcije metodom injekcije kojom se jednadžbe grupiraju i iz svake grupe jednadžbe odabere se jedna za koju se izračunava korekcija. Potom se dobivena korekcija preslika na čitavu grupu jednadžbi. U ovom radu koristit će se selektivni algebarski multigrid algoritam.

Matrica na gruboj razini, \mathbf{A}^C , definira se kao Galerkinova matrica:

$$\mathbf{A}^C = \mathbf{R}\mathbf{A}^F\mathbf{P}, \quad (3.28)$$

što je veoma bitno za svojstva multigrid matrice \mathbf{M} koja će biti navedena niže. Nadalje, matrica na gruboj razini \mathbf{A}^C rješava se direktnim metodama, kao što su *Gaussove eliminacije*, čime se dobiva točna vrijednost korekcije \mathbf{e}^C umjesto njezina aproksimacija. Raspisom matrice \mathbf{M} pomoću \mathbf{A}^C zapisane kao Galerkinova matrica dobiva se sljedeći izraz:

$$\mathbf{M} = \mathbf{P}(\mathbf{R}\mathbf{A}^F\mathbf{P})^{-1}\mathbf{R}\mathbf{A}^F, \quad (3.29)$$

koji ako se kvadrira nastaje:

$$\mathbf{M}^2 = \mathbf{P}(\mathbf{R}\mathbf{A}^F\mathbf{P})^{-1}\mathbf{R}\mathbf{A}^F\mathbf{P}(\mathbf{R}\mathbf{A}^F\mathbf{P})^{-1}\mathbf{R}\mathbf{A}^F = \mathbf{P}(\mathbf{R}\mathbf{A}^F\mathbf{P})^{-1}\mathbf{R}\mathbf{A}^F = \mathbf{M}. \quad (3.30)$$

Proučavajući svojstvene vektore matrice \mathbf{M} , na temelju izraza (3.30) može se zaključiti sljedeće:

$$\begin{cases} \mathbf{M}\mathbf{v} = \lambda\mathbf{v} \\ \mathbf{M}^2\mathbf{v} = \lambda^2\mathbf{v} \end{cases} \quad (3.31)$$

što daje:

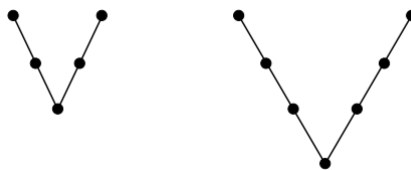
$$\begin{aligned} \mathbf{M}^2 &\equiv \mathbf{M} \\ \lambda\mathbf{v} &= \lambda^2\mathbf{v} \\ \lambda^2 - \lambda &= 0 \\ (\lambda - 1)\lambda &= 0. \end{aligned} \quad (3.32)$$

Iz navedenih izraza vidljivo je da postoje dvije svojstvene vrijednosti matrice \mathbf{M} , $\lambda_1 = 0$ i $\lambda_2 = 1$. Komponente greške sa svojstvenom vrijednosti $\lambda_2 = 1$ bit će potpuno

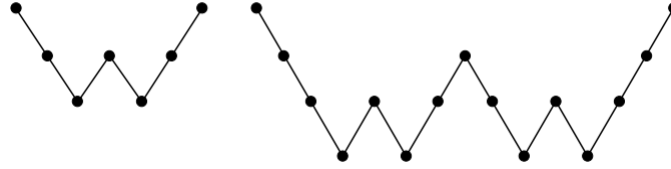
eliminirane, dok multigrid algoritam neće imati nikakvog utjecaja na komponente greške čija je svojstvena vrijednost $\lambda_1 = 0$. Upravo zbog ove činjenice nema smisla ponavljati multigrid V-ciklus s dvije razine više puta, već se za nadopunu multigrid algoritma s dvije razine koriste metode stacionarne točke. One zaglađuju rješenje na finoj razini prije izračunavanje reziduala, takozvani *pre-smoothing*, i rješenje nakon provedenog V-ciklusa multigrid algoritma, *post-smoothing*. Zbog njihovog djelovanja na smanjivanje određenih komponenti greške, dok ostale komponente ne povećavaju, metode stacionarne točke nazivaju se zaglađivači (eng. *smoothers*). Kao zaglađivač koristi se i ILU metoda [11] kojoj odgovaraju nešto drugačije postavke multigrid algoritma što će biti objašnjeno kasnije u radu. Matrica s primijenjenim *pre* i *post-smoothing*-om izgleda kako slijedi:

$$\underbrace{\mathbf{S}}_{\text{pre-smoothing}} \mathbf{M} \underbrace{\mathbf{S}}_{\text{post-smoothing}}, \quad (3.33)$$

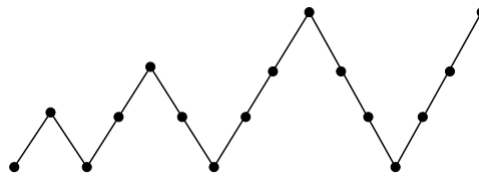
gdje \mathbf{S} označava *smoother* (zaglađivač). Primjena zaglađivača prije i nakon multigrid V-ciklusa može biti višekratna kako bi se što više reducirale visokofrekventne komponente greški. U sklopu ovog rada bit će ispitano koliko je *pre-smoothing* i *post-smoothing* procesa optimalno za konvergenciju rješenja i kakav utjecaj na rješenje ima svaki od odabranih algoritama zaglađivača. Iako nema smisla ponavljati multigrid V-ciklus s dvije razine, pokazano je da je za bržu konvergenciju rješenja poželjno stvaranje više grubih razina i prolazeći kroz njih ponavljati korake multigrid algoritma s dvije razine. Primjeri nekih od često korištenih multigrid ciklusa nalaze se na slikama 3.2, 3.3 i 3.4:



Slika 3.2: Multigrid V-ciklus s više grubih razina



Slika 3.3: Multigrid W-ciklus s više grubih razina



Slika 3.4: Cijeli multigrid ciklus

gdje je ideja da se dužim boravkom na grubim razinama dolazi do točnijih iznosa korekcije e^C .

3.5. Zaključak

U ovom poglavlju predstavljeni su algoritmi stacionarne točke za izračun aproksimacije rješenja linearnog sustava jednadžbi koji zbog prirode svojstvenih vrijednosti matrice nisu u mogućnosti eliminirati niskofrekventne komponente greške. Iz navedenog ograničenja algoritama proizlazi motivacija za korištenjem multigrid algoritma. Multigrid algoritam stvara grube razine matrice linearnog sustava \mathbf{A} na kojima izračunava grešku (korekciju) i potom ju vraća natrag na finu razinu matrice. Ovim postupkom multigrid eliminira visokofrekventne komponente greške, a niskofrekventne komponente pretvara u visokofrekventne koje se nakon primjene multigrid algoritma mogu ukloniti nekom od metoda stacionarne točke. Opisan je i ILU algoritam za rješavanje sustava linearnih jednadžbi koji se također može koristiti kao zaglađivač (*smoother*). Objasnjena je struktura multigrid matrice, osnovni V-ciklus algoritma te je dan uvid u najčešće korištene multigrid cikluse.

U sljedećem poglavlju bit će detaljno objašnjene postavke multigrid algoritma u *foam-extend*-u, njihova lokacija u rječniku te moguće varijante.

4 Postavke multigrid algoritma

4.1. Uvod

U prethodnom poglavlju detaljno su opisani motivacija, svrha i struktura multigrid algoritma popraćeni matematičkim dokazima o svojstvima multigrid matrice i posljedično sposobnostima algoritama stacionarne točke da uklone pojedine komponente greške.

Sljedeće poglavlje bavi se postavkama multigrid algoritma u *foam-extend*-u. Bit će objašnjeno gdje se postavke algoritma nalaze (u kojem rječniku), koje je sve funkcije algoritma moguće mijenjati i kako promjene postavki utječu na to kako algoritam operira nad matricom linearnog sustava \mathbf{A} .

4.2. Opis lokacije i promjena postavki multigrid algoritma

Kako bi se multigrid algoritam primijenio na neki problem, potrebno ga je *ugraditi* unutar rječnika *fvSolution* u *system* mapi. Unutar dijela rječnika gdje se postavljaju rješavači (eng. *solvers*), postavke multigrid algoritma upisuju se pod varijablu na čije se rješavanje matrice algoritam želi primijeniti. Primjer postavki rješavača za rješavanje polja tlaka prikazan je na slici 4.1:

```

solvers
{
    p
    {
        solver          amgSolver;
        cycle           V-cycle;

        policy          selectionAMG;

        nPreSweeps      2;
        nPostSweeps     2;
        minCoarseEqns   100;
        nMaxLevels      100;
        scale            off;
        smoother        GaussSeidel;

        minIter         0;
        maxIter         100;
        tolerance       1e-6;
        relTol          0;
    }
}

```

Slika 4.1: Postavke multigrid algoritma unutar rječnika *fvSolution*

Na slici 4.1 može se uočiti niz postavki koje će redom biti objašnjene u sljedećim potpoglavljima.

4.2.1. Odabir rješavača

Prva postavka unutar *fvSolution* rječnika jest odabir odgovarajućeg rješavača. Za pokretanje multigrid algoritma potrebno je pod *solver*, kao na slici 4.1, upisati *amgSolver* što je oznaka algebarskog multigrid rješavača.

4.2.2. Odabir multigrid ciklusa

Kako je navedeno u prethodnom poglavlju rada, postoji više varijanti multigrid ciklusa točnije, varijanti kako se multigrid algoritam kreće po grubim razinama. Na slici 3.2 prikazan je V-ciklus koji se sastoji od više grubih razina po kojima se algoritam kreće redom od najfinije do najgrublje razine i natrag (u obliku slova V). Da bi se odabrao V-ciklus u postavkama, potrebno je pod *cycle* u rječniku *fvSolution* 4.1 upisati *V-cycle*. Na slici 3.3 prikazan je W-ciklus kojim se multigrid algoritam kreće više puta po istim grubim razinama prije vraćanja na finu razinu (u obliku slova W). Ideja je da se duljim boravkom na grubim razinama postigne bolja korekcija na finoj razini i samim time brža

konvergencija. Za odabir W -ciklusa u postavkama, potrebno je pod *cycle* u navedenom rječniku upisati *W-cycle*.

4.2.3. Odabir načina formiranja matrice prolongacije i restrikcije

U poglavlju 3. navedena su dva načina kojima multigrad algoritam može kreirati matrice prolongacije i restrikcije s obzirom na koje algoritam nosi i različiti naziv. Matrice prolongacije i restrikcije koje se dobivaju metodom linearne interpolacije karakteriziraju selektivni algebarski multigrad algoritam, SAMG. Za odabir SAMG algoritma potrebno je pod *policy* u rječniku *fvSolution* upisati *selectionAMG*. Drugi algoritam je agregacijski algebarski multigrad algoritam, AAMG, koji izgrađuje matrice prolongacije i restrikcije metodom injekcije - jednadžbe se grupiraju i iz svake grupe jednadžbe odabere se jedna za koju se izračunava korekcija. Izračunata korekcija se potom primjenjuje na sve jednadžbe unutar grupe. Za odabir AAMG algoritma potrebno je pod *policy* u rječniku *fvSolution* upisati *AAMG*, a za odabir AAMG algoritma koji može grupirati samo dva kontrolna volumena odabire se *PAMG*. Postoje i druge opcije multigrad algoritma no one neće biti ovdje navedene.

4.2.4. Odabir algoritma zaglađivača i njegove primjene

U prethodnom poglavlju 3. objašnjeni su algoritmi stacionarne točke čiji su nedostaci postali motivacija za razvoj i korištenje multigrad algoritma. No, navedeni algoritmi imaju sposobnost eliminacije pojedinih komponenti greške bez utjecaja na druge komponente zbog čega se zovu zaglađivači (*smoothers*) i primjenjuju se prije samog multigrada za eliminaciju visokofrekventnih komponenti greške te nakon njega, kada multigrad pretvori niskofrekventne komponente u visokofrekventne [2]. Za definiranje zaglađivača potrebno je u rječniku *fvSolution* pod *smoother* upisati ime željenog algoritma. Na primjeru 4.1 kao zaglađivač odabran je Gauss-Seidel algoritam. U ovom radu u ulozi zaglađivača koristit će se još i ILU algoritam. Nakon odabira zaglađivača potrebno je odabrati koliko puta će se algoritam primijeniti prije (*pre-smoothing*) i nakon (*post-smoothing*) samog multigrad ciklusa. Utjecaj broja primjene zaglađivača prije i poslije multigrad ciklusa bit će testiran u ovom radu. Za odabir broja ponavljanja zaglađivanja prije multigrad ciklusa potrebno je u rječniku *fvSolution* upisati odgovarajući broj pod

$nPreSweeps$ dok je za odabir broja ponavljanja zaglađivanja prije multigrid ciklusa potrebno upisati odgovarajući broj pod $nPostSweeps$ kako je prikazano na primjeru 4.1.

4.2.5. Odabir minimalnog broja jednadžbi na gruboj razini

Osnova rada multigrid algoritma je stvaranje grubih matrica točnije, matrica sa smanjenim brojem jednadžbi u odnosu na finu matricu. Maksimalan broj jednadžbi u matrici na posljednjoj (najgrubljoj) razini ne može se ograničiti - on ovisi o načinu na koji algoritam određuje koje jednadžbe ostaju na gruboj razini. Matrica linearnog sustava \mathbf{A} sadrži informacije o međusobnom utjecaju kontrolnih volumena jedan na drugi - informacije koje su bitne za konvergenciju rješenja. Što je manji broj jednadžbi na najgrubljoj razini, to manje informacija iz fine razine ostaje zbog čega se postavkama može regulirati minimalan broj jednadžbi kako se ne bi izgubilo previše informacija u procesu redukcije jednadžbi (*coarsening*). Minimalan broj jednadžbi na gruboj razini zadaje se upisivanjem broja pod *minCoarseEqns* u rječniku *fvSolution*. Prilikom izgradnje grubih razina, multigrid algoritam stvara onoliko grubih razina koliko je moguće da uvjet minimalnog broja jednadžbi na najgrubljoj razini bude zadovoljen. Na primjer, ako je zadan minimalni broj jednadžbi 5, tada će posljednja gruba razina sadržavati 5 ili više jednadžbi, ali nikako manje.

4.2.6. Ograničavanje broja grubih razina

Još jedan način ograničavanja stvaranja grubih razina je eksplicitno zadavanje maksimalnog dopuštenog broja razina za razliku od prethodne postavke minimalnog broja jednadžbi koja indirektno utječe na broj grubih razina. Ograničavanje broja grubih razina postavlja se upisivanjem željenog broja pod *nMaxLevels* u rječniku *fvSolution*. Prilikom stvaranja grubih razina multigrid algoritam će stati kada dosegne zadani broj.

4.2.7. Ograničavanje broja iteracija

Prilikom rada, multigrid algoritam više puta prolazi kroz stvorene grube razine kako bi dosegao željenu točnost rješenja postavljenu od strane korisnika. Koliko će puta algoritam proći kroz svaku razinu u jednoj iteraciji ovisi o tipu ciklusa - V-ciklus gdje u svakoj iteraciji algoritam prolazi kroz svaku razinu jednom prikazano na slici 3.2 ili

W-ciklus gdje u jednoj iteraciji algoritam više puta prolazi kroz iste grube razine kao na slici 3.3 - ali se broj iteracija može ograničiti eksplicitno. Zadavanje minimalnog broja željenih iteracija radi se upisivanjem broja pod *minIter*, a zadavanje maksimalnog broja iteracija pod *maxIter* u rječniku *fvSolution*.

4.2.8. Postavke tolerancija

Na slici 4.1 u zadnja dva retka nalaze se tolerancija (*tolerance*) i relativna tolerancija (*relTol*). Tolerancija označava vrijednost finalnog reziduala, odnosno razlike $\mathbf{b} - \mathbf{Ax}$ gdje je \mathbf{x} rješenje sustava dobiveno u posljednjoj iteraciji, \mathbf{A} matrica linearnog sustava izgrađena u posljednjoj nelinearnoj iteraciji i \mathbf{b} vektor desne strane također izgrađen u posljednjoj nelinearnoj iteraciji. Zadaje se upisivanjem željenog broja pod *tolerance* u rječniku *fvSolution*. Relativna tolerancija je razlika finalnog i početnog reziduala gdje je početni rezidual razlika $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ u kojoj su matrica linearnog sustava \mathbf{A} i vektor desne strane \mathbf{b} izgrađeni u trenutnoj nelinearnoj iteraciji, a rješenje \mathbf{x} predstavlja rješenje sustava dobiveno u prethodnoj nelinearnoj iteraciji. Ona se također zadaje upisivanjem željenog broja pod *relTol* u rječniku *fvSolution*.

4.3. Promjena postavki unutar izvornog koda algoritma

Osim navedenih postavki, postoje i postavke koje korisnik ne može mijenjati unutar rječnika *fvSolution* već ih treba promijeniti u izvornom kodu algoritma i kompajlirati kako bi one bile trajno ugrađene u sam kod. To su postavke snage veza između jednadžbi i postavke poretka jednadžbi na gruboj razini matrice linearnog sustava.

4.3.1. Snaga veze između jednadžbi

Snaga veze između jednadžbi je koeficijent ε koji ima ulogu selekcije jednadžbi na finoj razini koje će biti zadržane na gruboj razini ili eliminirane u procesu *coarsening*-a. U svakom retku matrice \mathbf{A} odabire se koeficijent s najvećom apsolutnom vrijednosti i on se množi s ε . Dobivena vrijednost umnoška predstavlja graničnu vrijednost u izboru jednadžbi, odnosno ako je neki koeficijent u tom retku veći od dobivenog umnoška,

$|a_{ij}| > \max |a_{ij}| \cdot \varepsilon$, veza između promatrane jednadžbe i jednadžbe koju predstavlja taj koeficijent se smatra jakom vezom. Što jednadžba ima više jakih veza to ona ima više utjecaja u matrici. Jednadžbe se u procesu *coarsening*-a poredaju se po utjecajnosti i eliminacija jednadžbi kreće od najutjecajnije jednadžbe koja se zadržava na finoj razini, a sve jednadžbe kojima je ta jednadžba jaka veza se eliminiraju. Tada se kreće na sljedeću najutjecajnijiu jednadžbu koja nije eliminirana u prethodnom koraku i proces se nastavlja dok sve jednadžbe nisu sortirane kao grube ili fine jednadžbe. Jednadžbe se mogu poredati i drugačije što će biti opisano u sljedećem potpoglavlju. Snaga veze je broj $0 < \varepsilon < 1$, obično oko 0,25, a što je taj broj veći to će manje koeficijenata u pojedinom retku matrice zadovoljavati uvjet jake veze i proces eliminacije jednadžbi na grubom nivou će teći drugačije. U ovom radu bit će mijenjana snaga veze kako bi se utvrdio utjecaj promjene snage veze na rad multigrid algoritma.

4.3.2. Poredak jednadžbi na gruboj razini

Kako je navedeno u prethodnom potpoglavlju 4.3.1., jednadžbe u procesu *coarsening*-a mogu biti poredane prema utjecajnosti (broju jakih veza s drugim jednadžbama u matrici) što se naziva poretkom *strongest first*, odnosno poredak jednadžbi od najutjecajnijih prema najmanje utjecajnim. Druga mogućnost je zadržavanje poretka jednadžbi kao u finoj matrici, *fine order*, bez obzira na njihovu utjecajnost gdje onda *coarsening* kreće od prve jednadžbe u finoj matrici i matrica na gruboj razini izgleda potpuno drugačije. Prema teoriji objašnjenom u poglavlju 3., poredak jednadžbi *strongest first* trebao bi odgovarati zaglađivaču Gauss-Seidel, a *fine order* ILU zaglađivaču. Navedene tvrdnje bit će ispitane u ovom radu.

4.4. Zaključak

U ovom poglavlju prezentirane su postavke multigrid algoritma i njihove različite opcije u sklopu rječnika *fvSolution* u verziji *foam-extend* 4.0. Iznesena su objašnjenja utjecaja promjena postavki na stvaranje grubih razina matrice linearnog sustava \mathbf{A} i operiranja multigrid algoritma nad stvorenim razinama.

Sljedeće poglavlje bavi se odabranom geometrijom za testiranje multigrid algoritma. Na istoj geometriji bit će izrađene proračunske mreže različite gustoće i rasporeda kon-

trolnih volumena kako bi se istražili utjecaji različitih mreža na stvaranje grubih razina, konvergenciju rješenja i općenito na rad samog multigrad algoritma.

5 Geometrija i proračunska mreža

5.1. Uvod

U prethodnom poglavlju objašnjene su postavke multigrid algoritma koje će kasnije u radu biti mijenjane u svrhu proučavanja njihovih utjecaja na konvergenciju rješenja i formiranje matrice linearnog sustava \mathbf{A} .

Sljedeće poglavlje bavi se odabranom geometrijom i proračunskim mrežama izrađenim nad njom. Bit će opisani izgled i dimenzije domene te mreže različite gustoće (broja kontrolnih volumena). Nadalje, bit će dan i opis rubnih ploha domene s rubnim uvjetima postavljenim na rubnim plohama.

5.2. Geometrija

Odabrana geometrija za provedbu testiranja multigrid algoritma sastoji se od jedne cijevi i proširenja i izgleda kako slijedi na slici 5.1:



Slika 5.1: Odabrana geometrija s prikazanim točkama u x - y ravnini

Geometrija je izrađena pomoću *blockMeshDict* rječnika definirajući 8 točaka u ravnini x - y te sljedećih 8 točaka u ravnini paralelnoj s ravninom x - y udaljenom od nje za iznos 1. Prikazani slučaj je dvodimenzionalan - duljina geometrije po osi z je 1. Iako je geometrija jednostavna, ona sadrži bitne tipične probleme kada se promatra strujanje kao što su naglo proširenje u kojem se mogu stvarati vrtlozi i druge pojave te izobrazba strujanja po duljini cijevi. Također, bez obzira na jednostavnost geometrije, ona omogućava brojne varijacije što se tiče izrade proračunske mreže što će biti prikazano u sljedećim dijelovima ovog poglavlja.

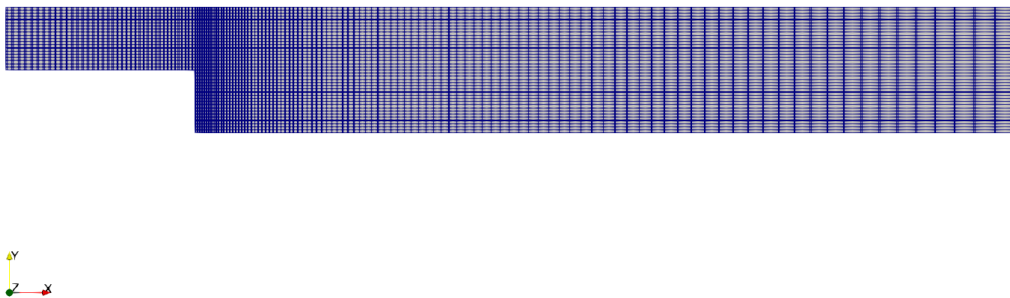
5.2.1. Proračunske mreže

Kao što je već spomenuto, u ovom radu izrađeno je više proračunskih mreža na istoj geometriji opisanoj u prethodnom dijelu poglavlja. Mreže se razlikuju prema broju kontrolnih volumena i njihovom obliku - kontrolni volumeni povećavaju svoje dimenzije u smjeru x i/ili y -osi ovisno o njihovom broju i gradaciji mreže. U svrhu ovog rada izrađene su tri heksaedarske mreže različitih gustoća na istoj geometriji - mreže s 4800, 19200 i 76800 kontrolnih volumena - jedna heksaedarska mreža na geometriji promijenjenih dimenzija - s 25520 kontrolnih volumena - te dvije mreže s kontrolnim volumenima u obliku tetraedra koje će biti opisane u nastavku.

Mreža s 4800 kontrolnih volumena

Najrjeđa mreža korištena u ovom radu jest mreža s 4800 kontrolnih volumena sa sljedećim osobitostima:

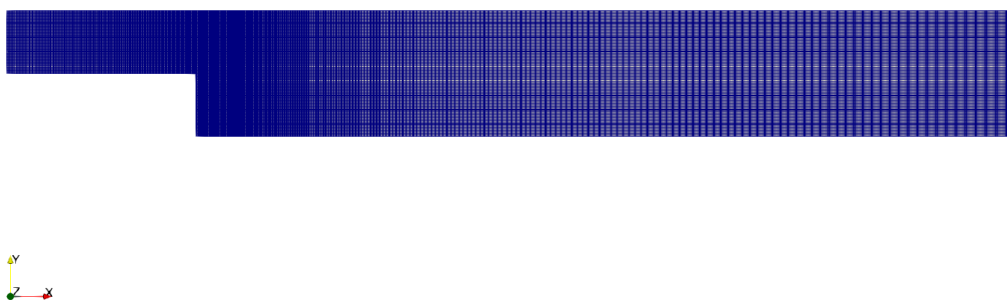
izgrađena je pomoću rječnika *blockMeshDict* izgrađujući blokove pomoću naredbe *hex* i zadajući željeni broj kontrolnih volumena u smjeru osi x i y (izgrađuje se dvodimenzionalna proračunska mreža). Na slici 5.2 prikazane su naredbe u rječniku *blockMeshDict* gdje su u prvoj zagradi iza naredbe *hex* napisani nazivi točaka koje trebaju biti u bloku, u drugoj zagradi broj kontrolnih volumena u tom bloku u smjeru osi x , y i z , potom naredba *simpleGrading* kojom se određuje željena gradacija mreže za svaki blok u smjeru sve 3 osi. Gradacija mreže označava zgušnjavanje ili razrjeđivanje mreže u smjeru pojedinih osi za svaki blok. Mreža je progušćena na izlazu iz manje cijevi i na proširenju pošto se tamo očekuju najznačajnije promjene u strujanju. Na izlazu iz cijevi većeg promjera (na izlazu iz čitave geometrije) kontrolni volumeni su većeg volumena i izduženi u smjeru osi x jer se na tom mjestu očekuje već izobraženo strujanje bez značajnih promjena.



Slika 5.2: Proračunska mreža s 4800 kontrolnih volumena

Mreža s 19200 kontrolnih volumena

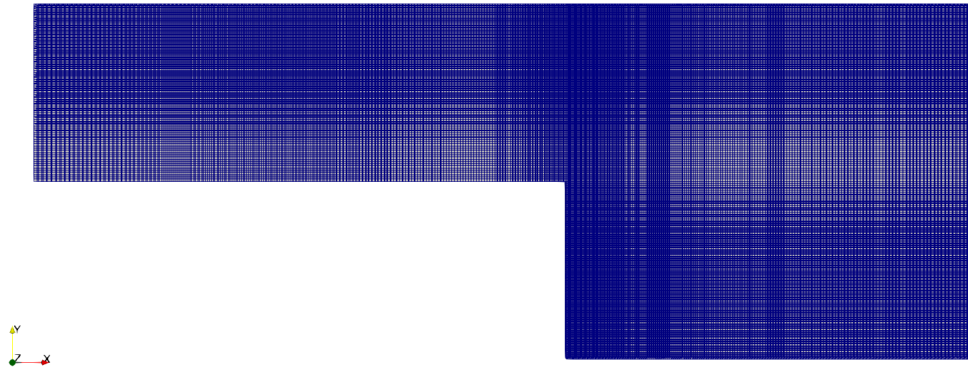
Sljedeća mreža izgrađena je na isti način kao i prethodna, pomoću rječnika *blockMeshDict* i istih naredbi osim što je u drugoj zagradi nakon naredbe *hex* broj kontrolnih volumena udvostručen u smjeru osi x i y u odnosu na prethodnu mrežu od 4800 kontrolnih volumena. Postavka gradacije je ista kao u prethodnoj mreži te je mreža progušćena na istim mjestima kao i prethodna. Nova mreža sastoji se od 19200 kontrolnih volumena i prikazana je na slici 5.3:



Slika 5.3: Proračunska mreža s 19200 kontrolnih volumena

Mreža sa 76800 kontrolnih volumena

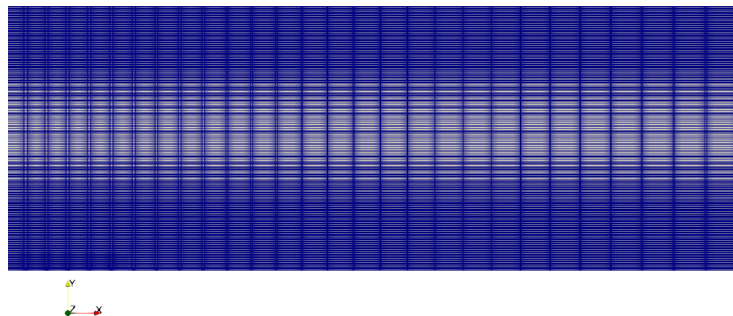
Sljedeća mreža izgrađena je na isti način kao prethodne dvije, s ponovnim udvostručavanjem broja kontrolnih volumena u smjeru osi x i y u odnosu na prethodnu mrežu od 19200 kontrolnih volumena. Nova mreža sastoji se od 76800 kontrolnih volumena te je postavka gradacije ista kao u prethodne dvije mreže. Progušćivanje mreže je ponovo na izlazu ic cijevi manjeg promjera te na proširenju. Nova mreža je zbog broja kontrolnih volumena veoma gusta te se u omjeru u kojem su prikazane prethodne dvije mreže ne razaznaju kontrolni volumeni zbog čega je na slici 5.4 prikazan samo dio mreže:



Slika 5.4: Dio proračunske mreže s 19200 kontrolnih volumena

Mreža s 25520 kontrolnih volumena

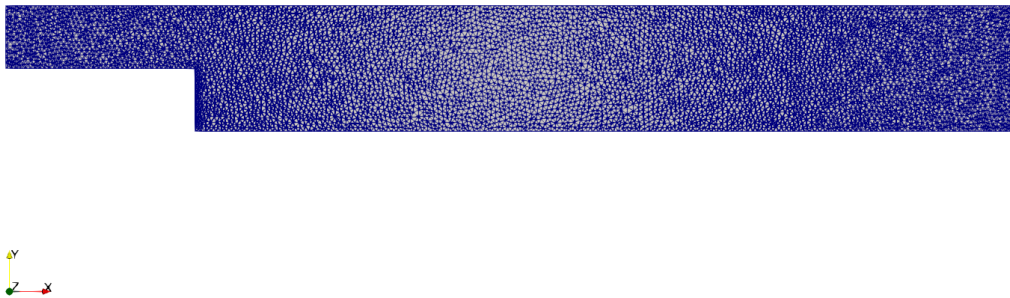
Mreža od 25520 kontrolnih volumena je mreža prilikom čije je izrade zadržana ista geometrija, ali je broj kontrolnih volumena u smjeru osi x smanjen za trećinu u svakom bloku u odnosu na mrežu sa 76800 kontrolnih volumena. Time je dobivena rjeđa mreža i kontrolni volumeni koji su izduženi u smjeru osi x , posebice na izlazu iz cijevi većeg promjera. Svrha izrade ove mreže bi će pojašnjena u kasnijim dijelovima rada. Izduženi kontrolni volumeni na izlazu iz cijevi većeg promjera prikazani su na slici 5.5:



Slika 5.5: Dio proračunske mreže (izlaz) s 25520 kontrolnih volumena

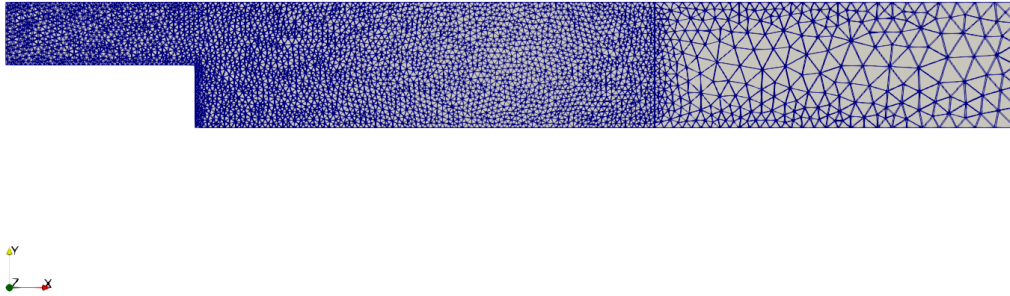
Tetraedarske mreže

U sklopu ovog rada pojedine postavke AMG algoritma i algoritma zaglađivača bit će testirane i na mrežama koje umjesto heksaedara kao kontrolnih volumena imaju tetraedre. Prva takva mreža jest mreža s istim mjestima zgušnjavanja kontrolnih volumena kao dosad prikazane mreže, prikazana na slici 5.6, koja ima 11046 kontrolnih volumena u obliku tetraedra.



Slika 5.6: Proračunska mreža s 11046 kontrolnih volumena u obliku tetraedara

Druga tetraedarska mreža izrađena u ovom radu jest mreža s povećanim kontrolnim volumenima na izlazu iz cijevi većeg promjera u smjeru osi x , poput mreže 5.5 osim što umjesto heksaedara sadrži tetraedre. Mreža se sastoji od 6228 kontrolnih volumena i prikazana je na slici 5.7.



Slika 5.7: Proračunska mreža s 6228 kontrolnih volumena u obliku tetraedara

5.2.2. Rubne plohe geometrije i rubni uvjeti

Za provedbu simulacije u *foam-extend*-u potrebno je odrediti rubne plohe ili grupe rubnih ploha preko kojih model komunicira s vanjskim svijetom. Na tim plohamo postaviti će se rubni uvjeti koji će opisivati komunikaciju između modela i njegove okoline na svakoj plohi njihovog dodira.

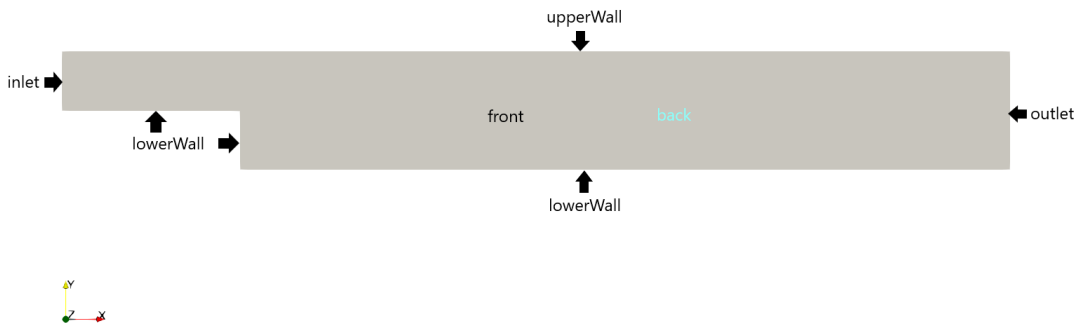
Rubne plohe

Bitno je uočiti da je model dvodimenzionalan stoga su sve plohe u modelu kvadratnog oblika, čak i one u z - y ravnini (kao da je model presječen ravninom x - y). Promatrajući geometriju sa slike 5.1, mogu se uočiti sljedeće plohe preko kojih model komunicira s okolinom:

- ulaz u cijev manjeg promjera - označena kao *inlet*
- izlaz iz cijevi većeg promjera - označena kao *outlet*
- gornja ploha - označena kao *upperWall*
- donja ploha - označena kao *lowerWall*

- prednja i stražnja ploha - označene kao *frontAndBack*

Oznake ploha vidljive su na sljedećoj slici 5.8



Slika 5.8: Oznake ploha/grupa ploha odabrane geometrije

Rubni uvjeti

Nakon definiranja rubnih ploha, na njima se postavljaju rubni uvjeti koji govore o načinu komunikacije modela s okolinom preko rubnih ploha. Rubni uvjeti postavljaju se za brzinu \mathbf{U} , tlak p , kinetičku energiju turbulencije k , disipaciju kinetičke energije turbulencije ε i turbulentnu viskoznost ν .

Postoji više vrsta rubnih uvjeta koji će ovdje biti ukratko opisani. *Dirichlet* rubni uvjet je rubni uvjet fiksne vrijednosti (*fixedValue*) kojim se zadaje konstantna vrijednost neke veličine na rubnoj plohi. Sljedeći rubni uvjet je *Neumann* rubni uvjet kojim se zadaje nulti gradijent (*zeroGradient*) neke vrijednosti na rubnoj plohi. Zadavanjem *Neumann*-ovog rubnog uvjeta za neku veličinu vrijednost te veličine će se uzimati iz unutrašnjosti domene i preslikati na rubnu plohu. Za veličine koje se odnose na modeliranje turbulencije (k, ε i ν) postavljaju se zidne funkcije (*WallFunction*) koje služe modeliranju strujanja u blizini zida i neće biti detaljno objašnjene u ovom radu. Navedeni rubni uvjeti postavljaju se za sve gore navedene veličine od kojih se rubni uvjeti za brzinu \mathbf{U} , kinetičku energiju turbulencije k , disipaciju kinetičke energije turbulencije ε i turbulentnu viskoznost ν ne mijenjaju u radu, dok će rubni uvjeti za tlak p biti varirani, ali

samo na rubnoj plohi *outlet* ovisno o slučajevima koji će se testirati. S obzirom na to da je ovaj model dvodimenzionalan, na rubne plohe *frontAndBack* se postavlja rubni uvjet *empty* koji označava da se na tim plohama (i u tim smjerovima) ne rješavaju sustavi jednačbi. U tablici 5.1 prikazani su rubni uvjeti na plohama/grupama ploha sa slike 5.8:

	Rubni uvjeti				
Rubna ploha/grupa	$\mathbf{U} [m/s]$	$p [m^2/s^2]$	$k [m^2/s^2]$	$\varepsilon [m^2/s^3]$	$\nu [m^2/s]$
<i>inlet</i>	fixedValue; uniform (1 0 0);	zeroGradient;	fixedValue; uniform 0.00135;	fixedValue; uniform 0.0001;	calculated; uniform 0;
<i>outlet</i>	zeroGradient;	zeroGradient; or fixedValue; uniform (0 0 0);	zeroGradient;	zeroGradient	calculated; uniform 0;
<i>upperWall</i>	fixedValue; uniform (0 0 0);	zeroGradient;	kqRWallFunction; uniform 0.00135;	epsilonWallFunction; uniform 0;	nutWallFunction; uniform 0;
<i>lowerWall</i>	fixedValue; uniform (0 0 0);	zeroGradient;	kqRWallFunction; uniform 0.00135;	epsilonWallFunction; uniform 0;	nutWallFunction; uniform 0;
<i>frontAndBack</i>	empty;	empty;	empty;	empty;	empty;

Tablica 5.1: Tablica rubnih uvjeta

5.3. Zaključak

U ovom poglavlju opisana je geometrija odabranog modela izrađena pomoću *blockMeshDict* rječnika u *foam-extend-u*. Opisana je izrada mreža različite gustoće nad odabranom geometrijom te dan njihov pojedinačni prikaz. Nadalje, opisane su i rubne plohe modela koje predstavljaju dodirne plohe modela i njegove okoline te je dan detaljan opis rubnih uvjeta postavljenih na definiranim rubnim plohama, odnosno grupama rubnih ploha.

Sljedeće poglavlje dat će pregled dobivenih rezultata koristeći postavke multigrid algoritma odabrane za testiranje, varirane u različitim slučajevima i na opisanim mrežama različite gustoće i oblika kontrolnih volumena, kao i analizu rezultata i donesene zaključke.

6 Rezultati

6.1. Uvod

Prethodno poglavlje bavi se opisom odabrane geometrije i odgovarajućih proračunskih mreža. Kreirane su mreže različite gustoće (broja kontrolnih volumena) i oblika kontrolnih volumena (izduženost stranica u pojedinim smjerovima) na kojima će se provoditi testiranja različitih postavki multigrid algoritma. Također, opisane su i rubne plohe geometrije, odnosno plohe preko kojih model komunicira s okolinom, te navedeni rubni uvjeti postavljeni na pojedinim rubnim plohama.

U ovom poglavlju predstavljene su rezultati dobiveni nizom numeričkih simulacija provedenih na različitim mrežama varirajući postavke multigrid algoritma u *fvSolution* rječniku i u izvornom kodu algoritma. Dobiveni rezultati će biti opisani, prikazani tablično i na nekim mjestima grafički te analizirani s obzirom na teorijske tvrdnje i ostale dobivene rezultate. Primarni interes provođenja numeričkih simulacija je analiza utjecaja promjene postavki multigrid algoritma na konvergenciju reziduala, stvaranje grubih razina matrice, broja dijagonalno dominantnih jednadžbi, i slično, stoga prikaz fizikalnog rješenja polja tlaka i brzine nije u fokusu.

Poglavlje je podijeljeno na potpoglavlja s obzirom na promjenu određenih postavki multigrid algoritma radi lakše usporedbe rezultata i preglednosti rada.

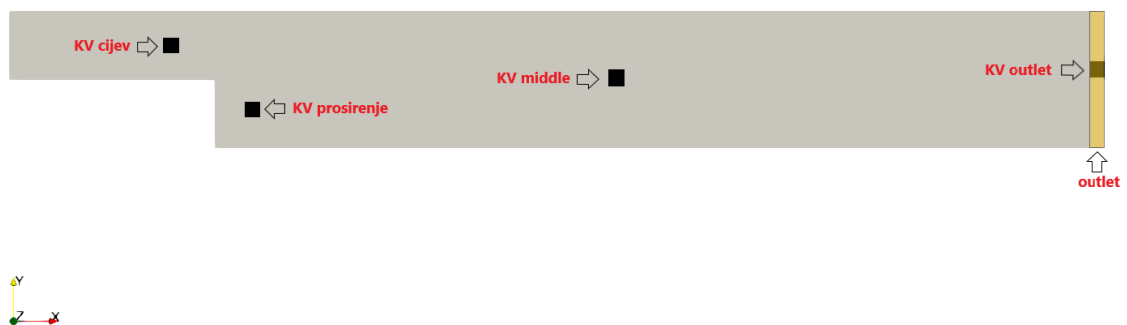
6.2. Oznake

U ovom poglavlju za prikaz rezultata koristit će se određene oznake koje je potrebno uvesti na samom početku. Oznake se odnose na specifične lokacije kontrolnih volu-

mena ili granice u/na kojima je zadana fiksna vrijednost tlaka (eng. *pressure boundary condition*). Korištene oznake su sljedeće:

- zadavanje tlaka na čitavoj izlaznoj rubnoj plohi - označeno sa slučaj outlet,
- zadavanje tlaka u jednom kontrolnom volumenu na izlaznoj rubnoj plohi (uz rubni uvjet *zeroGradient* na izlazu) - označeno kao slučaj KV outlet,
- zadavanje tlaka u jednom kontrolnom volumenu u sredini modela (uz rubni uvjet *zeroGradient* na izlazu) - označeno kao slučaj KV middle,
- zadavanje tlaka u jednom kontrolnom volumenu na proširenju cijevi (uz rubni uvjet *zeroGradient* na izlazu) - označeno kao slučaj KV proširenje,
- zadavanje tlaka u jednom kontrolnom volumenu u cijevi manjeg promjera (uz rubni uvjet *zeroGradient* na izlazu) - označeno kao slučaj KV cijev.

Odabrani položaji kontrolnih volumena vidljive su na slici 6.1:



Slika 6.1: Oznake položaja kontrolnih volumena i granice za zadavanje vrijednosti tlaka

6.3. Poredak jednadžbi

Kako je opisano u poglavlju 4., poredak jednadžbi na gruboj razini može biti prema utjecajnosti jednadžbi (koliko jakih veza ima pojedina jednadžba), *strongest first*, ili može biti isti kao u finoj matici neovisno o utjecajnosti jednadžbi, *fine order*. U ovom potpoglavlju testirana je postavka poretka jednadžbi na gruboj razini za zaglađivače Gauss-Seidel i ILU kako bi se provjerila teorijska tvrdnja da poredak jednadžbi *strongest first* odgovara zaglađivaču Gauss-Seidel, a *fine order* ILU zaglađivaču.

Postavke rubnih uvjeta navedene su u poglavlju 4.. Variran je položaj jednog rubnog uvjeta, a to je veličina tlaka zadana na rubnoj plohi *outlet* gdje je u jednoj varijanti zadana vrijednost tlaka 0 na čitavoj izlaznoj plohi dok je u drugoj varijanti na izlaznoj plohi zadan rubni uvjet *zeroGradient*, a fiksna vrijednost tlaka zadana je u jednom kontrolnom volumenu s variranjem položaja unutar mreže za različite slučajeve. Vrijednost tlaka u jednom kontrolnom volumenu zadaje se upisivanjem naziva kontrolnog volumena (njegov brojčani identitet) i iznosa tlaka unutar rječnika *fvSolution* pod postavkama SIMPLE algoritma. Ostali rubni uvjeti prikazani u tablici 5.1 nisu mijenjani.

6.3.1. *strongest first* poredak

Prvi testirani poredak jednadžbi jest poredak *strongest first* sa zaglađivačem Gauss-Seidel uz sljedeće postavke multigrid algoritma (opisanim u poglavlju 4.):

```

solvers
{
  p
  {
    solver          amgSolver;
    cycle           V-cycle;

    policy          selectionAMG;

    nPreSweeps     2;
    nPostSweeps    2;
    minCoarseEqns  5;
    nMaxLevels     100;
    scale           off;
    smoother       GaussSeidel;

    minIter        0;
    maxIter        100;
    tolerance      1e-6;
    relTol         0;
  }
}

```

Slika 6.2: Prikaz postavki multigrid algoritma unutar rječnika *fvSolution*

gdje su sve postavke fiksne tijekom više provedenih simulacija osim postavke minimalnog broja jednadžbi na gruboj razini, *minCoarseEqns*, za koju su odabrane tri varijacije broja jednadžbi - 5, 20 i 100. Simulacije su provedene na mrežama s 4800, 19200 i 76800 kontrolnih volumena opisanim u poglavlju 5. i rezultati su ispisani u 10. nelinearnoj iteraciji za sve tri mreže kako bi se usporedili rezultati na početku svake simulacije (još daleko od iskonvergirano rješenja za sve tri mreže). Mijenjan je i položaj kontrolnog volumena u kojem se zadaje fiksna vrijednost tlaka kako bi se proučio utjecaj zadavanja rubnog uvjeta na konvergenciju rješenja.

Promatrani rezultati su početni i finalni rezidual, broj kreiranih grubih razina matrice, broj iteracija multigrid algoritma i kumulativna greška na kraju nelinearne iteracije. Svaki od navedenih rezultata promatran je za sve tri mreže, za svaku od tri postavke minimalnog broja jednadžbi na gruboj razini i za svaki gore navedeni položaj kontrolnog volumena u kojoj je zadana vrijednost tlaka. U tablicama 6.1, 6.2 i 6.3 prikazani su dobiveni rezultati.

min coarse equations		5				
pressure boundary condition		outlet	KV outlet	KV middle	KV prosirenje	KV cijev
4800 cells	initial residual	8.06E-01	8.71E-01	8.60E-01	8.78E-01	8.49E-01
	final residual	8.97E-07	7.54E-07	7.97E-07	7.32E-07	9.76E-07
	number of coarse levels	6	7	7	6	6
	number of AMG iterations	18	21	26	23	19
	cumulative continuity error	-3.93E-04	-3.56E-15	2.12E-15	-7.91E-16	-8.42E-15
19200 cells	initial residual	9.35E-01	9.40E-01	9.39E-01	9.41E-01	9.41E-01
	final residual	9.02E-07	9.21E-07	9.61E-07	9.49E-07	9.63E-07
	number of coarse levels	7	8	7	8	8
	number of AMG iterations	21	26	18	27	19
	cumulative continuity error	-1.86E-02	-7.81E-13	-6.08E-13	-7.92E-14	-2.65E-13
76800 cells	initial residual	2.03E-01	1.09E-01	6.44E-01	8.97E-01	8.91E-01
	final residual	9.61E-07	8.49E-07	8.23E-07	9.88E-07	8.35E-07
	number of coarse levels	9	8	9	8	9
	number of AMG iterations	42	22	28	29	27
	cumulative continuity error	-5.98E-02	8.42E-13	1.53E-12	-1.43E-12	-6.08E-13

Tablica 6.1: Tablica rezultata simulacija s minimalnim brojem jednadžbi na gruboj razini 5

U tablici 6.1 može se uočiti da se ovisno o položaju kontrolnog volumena u kojem je zadana vrijednost tlaka javljaju različiti rezultati. Dok su općenito početni (eng. *initial*) i krajnji (eng. *final*) rezidual za sve tri mreže i sve pozicije kontrolnih volumena veoma slični, razlika je uočljiva kod kumulativne greške kontinuiteta (eng. *cumulative continuity error*) koja predstavlja prosječnu razliku protoka koji ulazi u kontrolni volumen i izlazi iz njega. Vidljivo je da je kumulativna greška kontinuiteta za zadanu fiksnu vrijednost tlaka na izlaznoj plohi (outlet) nekoliko redova veličine veća od kumulativne greške kontinuiteta za vrijednost tlaka zadanu u jednom kontrolnom volumenu na različitim mjestima. Primijećena odstupanja za kumulativnu grešku kontinuiteta vidljiva su za sve tri gustoće mreže i ponavljaju se za minimalni broj jednadžbi na gruboj razini 20 (tablica 6.2) i 100 (tablica 6.3). Broj kreiranih grubih razina (eng. *number of coarse equations*) i broj iteracija multigrad algoritma (eng. *number of AMG iterations*) ne pokazuju nikakav specifični uzorak ovisno o promjeni mjesta kontrolnog volumena sa zadanim tlakom. Može se još zamijetiti da se broj kreiranih grubih razina povećava s povećanjem gustoće mreže što je i očekivano s obzirom na to da gušća mreža ima više kontrolnih volumena i samim time više redaka matrice linearnog sustava stoga se proces *coarsening*-a ponav-

lja više puta do dostizanja minimalno mogućeg broja jednadžbi (prema korisnikovim postavkama).

min coarse equations		20				
pressure boundary condition		outlet	KV outlet	KV middle	KV prosirenje	KV cijev
4800 cells	initial residual	8.06E-01	8.71E-01	8.60E-01	8.78E-01	8.49E-01
	final residual	8.35E-07	8.62E-07	8.81E-07	7.00E-07	7.96E-07
	number of coarse levels	5	6	5	6	6
	number of AMG iterations	17	20	17	20	20
	cumulative continuity error	-8.28E-06	-2.64E-15	-4.53E-15	1.38E-14	1.24E-15
19200 cells	initial residual	9.34E-01	9.40E-01	9.39E-01	9.41E-01	9.41E-01
	final residual	8.47E-07	7.64E-07	7.41E-07	9.52E-07	8.37E-07
	number of coarse levels	6	6	6	6	6
	number of AMG iterations	16	19	21	20	19
	cumulative continuity error	-4.15E-05	-1.86E-13	1.92E-13	-2.16E-12	4.80E-13
76800 cells	initial residual	3.63E-03	1.10E-01	6.44E-01	8.97E-01	8.51E-01
	final residual	6.97E-07	7.48E-07	8.16E-07	7.27E-07	7.78E-07
	number of coarse levels	8	7	7	7	7
	number of AMG iterations	10	19	19	20	20
	cumulative continuity error	1.13E+00	-1.33E-12	3.98E-12	6.44E-13	-4.65E-13

Tablica 6.2: Tablica rezultata simulacija s minimalnim brojem jednadžbi na gruboj razini 20

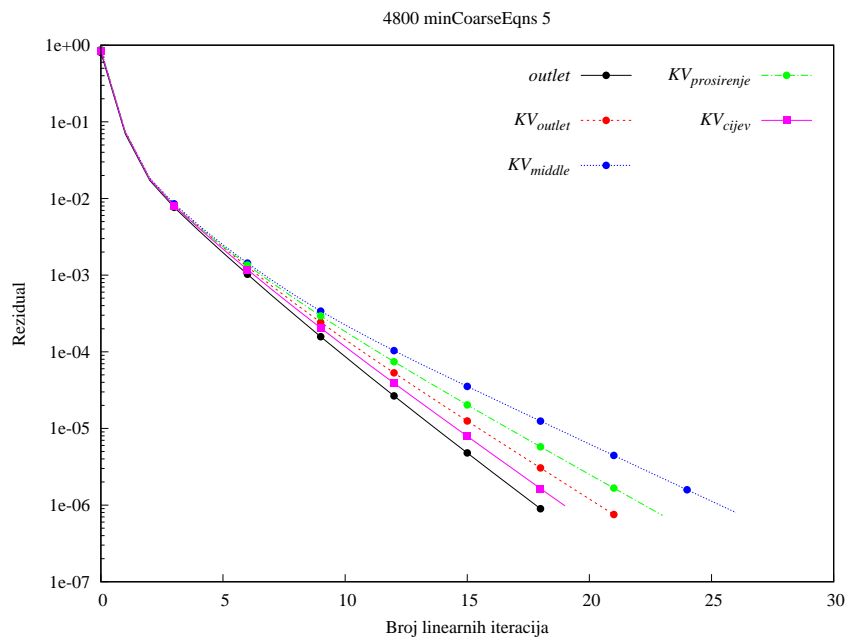
Proučavajući tablicu 6.2 može se uočiti da je broj grubih razina manji u odnosu na prethodnu tablicu što je i logično s obzirom da postavku minimalnog broja jednadžbi na gruboj razini 20 koja ograničava kreiranje grubih razina s manjim brojem jednadžbi od zadanog. Osvrne li se ponovo na kumulativnu grešku kontinuiteta, može se uočiti njezino smanjenje za jedan do dva reda veličine za varijantu outlet za mreže s 4800 i 19200 kontrolnih volumena u usporedbi s mrežama iste gustoće za postavku minimalnog broja jednadžbi na gruboj razini 5.

min coarse equations		100				
pressure boundary condition		outlet	KV outlet	KV middle	KV prosirenje	KV cijev
4800 cells	initial residual	8.06E-01	8.71E-01	8.60E-01	8.78E-01	8.49E-01
	final residual	6.76E-07	7.10E-07	7.02E-07	7.15E-07	6.86E-07
	number of coarse levels	4	4	4	4	4
	number of AMG iterations	14	14	14	14	14
	cumulative continuity error	1.43E-06	-2.44E-14	-1.57E-14	1.03E-14	-2.20E-14
19200 cells	initial residual	9.34E-01	9.40E-01	9.39E-01	9.41E-01	9.41E-01
	final residual	5.87E-07	7.85E-07	7.75E-07	9.41E-07	5.61E-07
	number of coarse levels	5	5	5	5	5
	number of AMG iterations	15	13	13	14	14
	cumulative continuity error	-1.15E-05	3.18E-13	1.06E-12	-1.85E-13	-2.15E-14
76800 cells	initial residual	2.03E-01	1.09E-01	6.44E-01	8.97E-01	8.91E-01
	final residual	9.71E-07	6.70E-07	8.64E-07	7.12E-07	7.39E-07
	number of coarse levels	6	6	6	6	6
	number of AMG iterations	16	14	17	16	18
	cumulative continuity error	-7.41E-04	-1.83E-12	1.75E-12	3.94E-13	1.87E-12

Tablica 6.3: Tablica rezultata simulacija s minimalnim brojem jednadžbi na gruboj razini 100

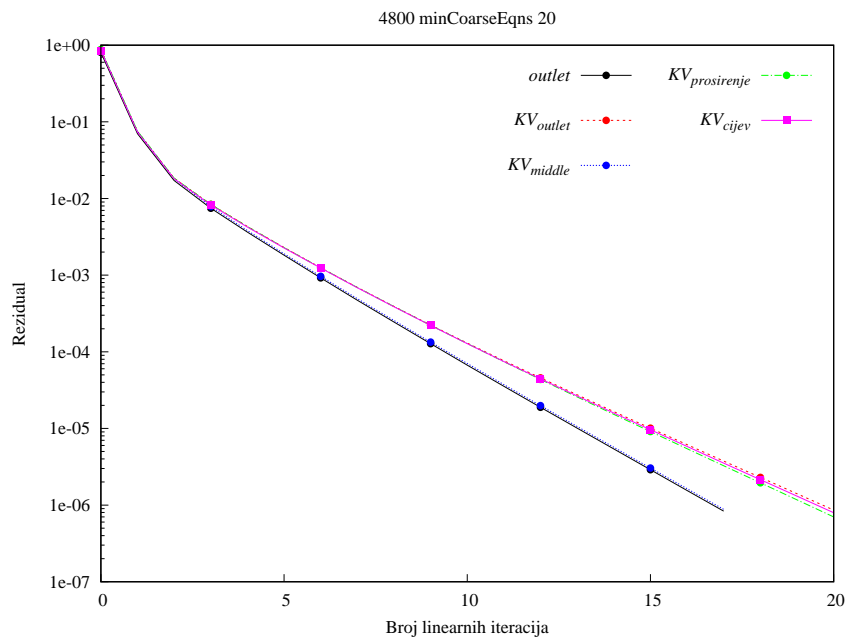
U tablici 6.3 uočava se smanjivanje broja grubih razina u odnosu na prethodna dva slučaja zbog povećanja zahtjeva za minimalan broj jednadžbi na gruboj razini koji sada iznosi 100. Ponovo se može uočiti smanjenje kumulativne greške kontinuiteta za varijantu outlet u odnosu na prethodni slučaj 6.2, ovaj put kod mreže sa 76800 kontrolnih volumena. Kod ostalih varijanti outlet i zadanog tlaka u jednom kontrolnom volumenu nema primjetnih značajnijih promjena.

Pošto je iz tabličnog prikaza rezultata veoma teško zaključiti o brzini konvergencije reziduala linearnog rješavača, izrađeni su grafički prikazi u 10. nelinearnoj iteraciji (isto kao u prikazanim tablicama). Na sljedećim grafovima prikazana je konvergencija reziduala za svaku od mreža navedenih u tablicama 6.1, 6.2 i 6.3 i za svaku postavku minimalnog broja jednadžbi na gruboj razini.



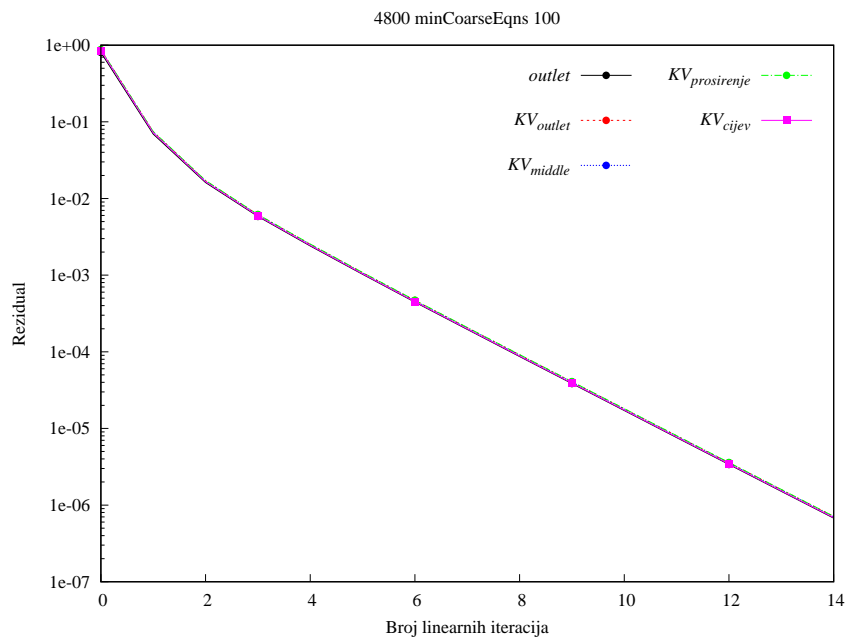
Slika 6.3: Konvergencija reziduala linearnog rješavača - mreža od 4800 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 5

Graf na slici 6.3 prikazuje konvergenciju reziduala za mrežu od 4800 kontrolnih volumena sa zadanim minimalnim brojem jednažbi na gruboj razini 5. Uočava se nekoliko različitih linija koje predstavljaju konvergenciju reziduala ovisno o lokaciji zadavanja tlaka u modelu (koje su opisane na početku ovog poglavlja). Vidljivo je da u prvim linearnim iteracijama konvergencija reziduala uopće ne ovisi o položaju kontrolnog volumena u kojem se zadaje tlak, dok se u kasnijim iteracijama, s približavanjem zadanoj toleranciji, brzina konvergencije počinje razilaziti za različite slučajeve.



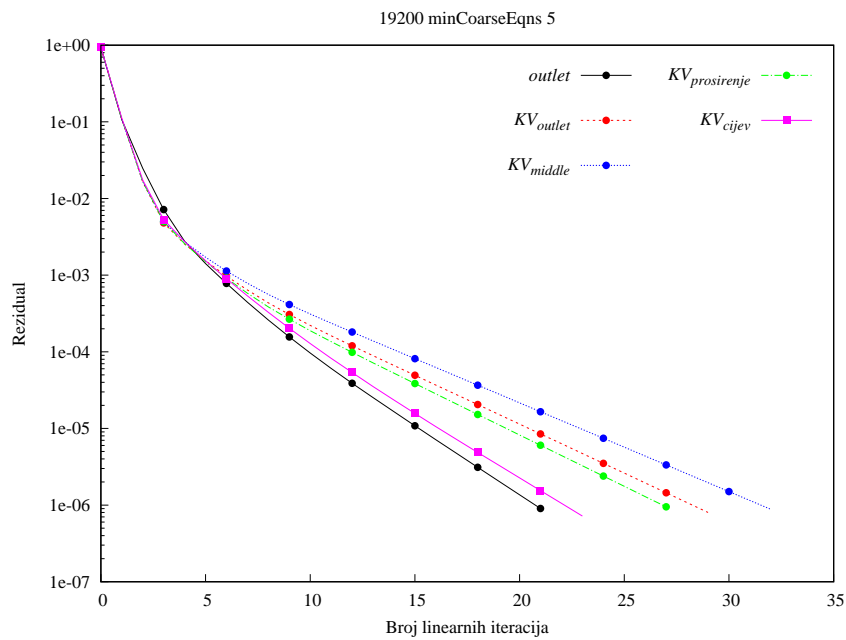
Slika 6.4: Konvergencija reziduala linearnog rješavača - mreža od 4800 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 20

Na slici 6.4 prikazani su rezultati dobiveni na istoj mreži, ali s postavkom minimalnog broja jednažbi na gruboj razini 20. Može se uočiti da u ovom slučaju također ne postoji razlika u konvergenciji u prvim iteracijama, ali i da se konvergencija reziduala u kasnijim iteracijama preklapa za određene slučajeve. Također, vidljivo je smanjenje broja iteracija u odnosu na minimalan broj jednažbi na gruboj razini 5.



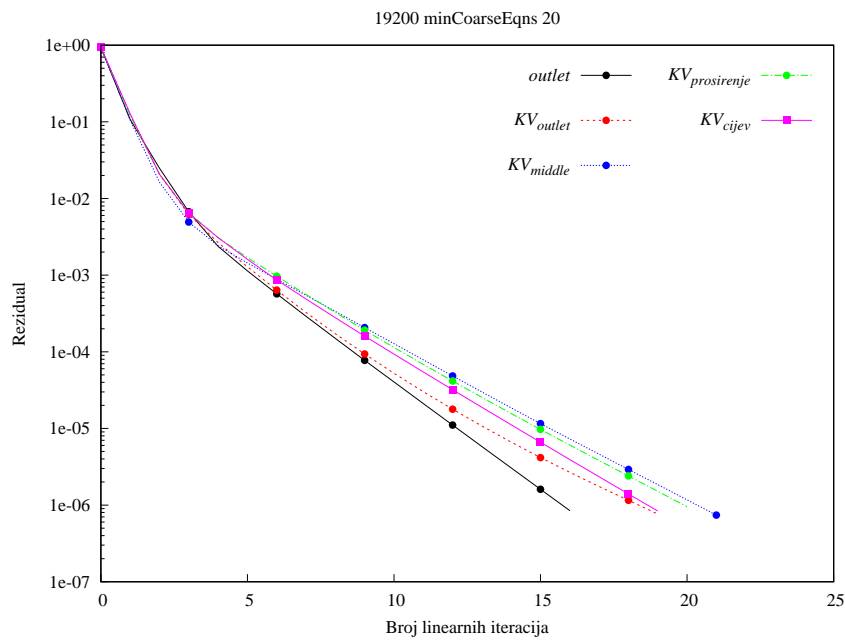
Slika 6.5: Konvergencija reziduala linearnog rješavača - mreža od 4800 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 100

Slika 6.5 prikazuje treći slučaj na mreži od 4800 kontrolnih volumena gdje je minimalni broj jednažbi na gruboj razini 100. Ovdje se može uočiti kako se razlika konvergencije reziduala ovisno o položaju zadavanja vrijednosti tlaka potpuno izgubila, odnosno brzina konvergencije je ista za svih 5 slučajeva. Također, uočava se daljnje smanjenje broja iteracija linearnog rješavača.



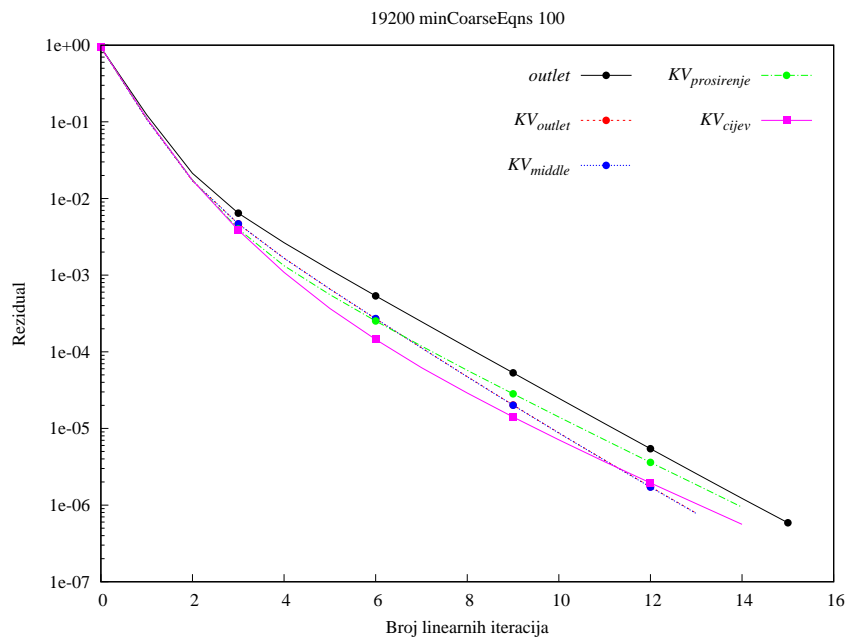
Slika 6.6: Konvergencija reziduala linearnog rješavača - mreža od 19200 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 5

Na slici 6.6 prikazani su rezultati dobiveni na mreži od 19200 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 5 gdje je, kao i na slici 6.3, vidljivo razilaženje brzine konvergencije pojedinih slučajeva nakon prvih nekoliko linearnih iteracija s preklapanjem u prvim iteracijama. Broj linearnih iteracija je značajno veći u odnosu na prethodnu mrežu s istim postavkama zbog većeg broja kontrolnih volumena (većeg broja jednažbi u matrici linearnog sustava \mathbf{A}).



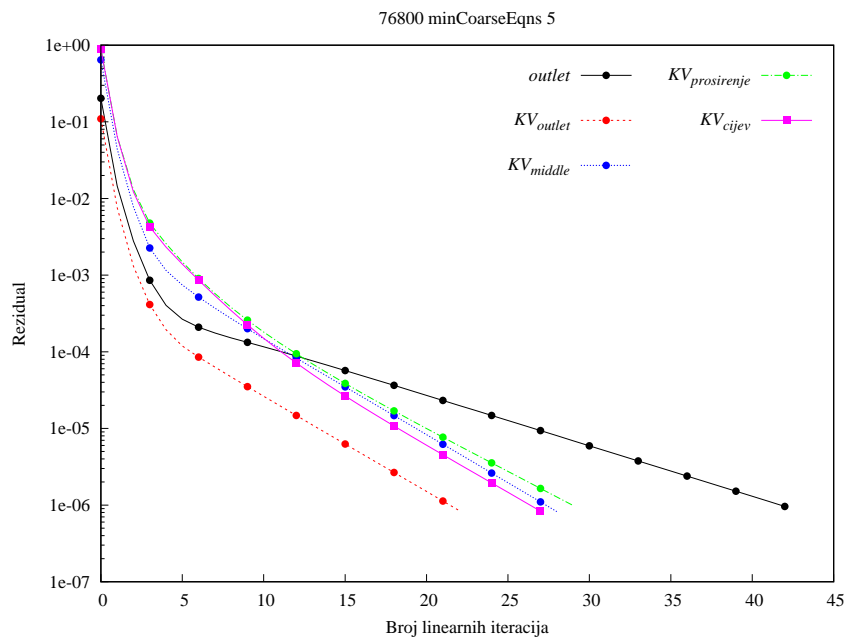
Slika 6.7: Konvergencija reziduala linearnog rješavača - mreža od 19200 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 20

Usporedi li se sliku 6.7 sa slikom 6.4, kod mreže od 19200 kontrolnih volumena može se uočiti veća razlika u brzini konvergencije za različite slučajeve zadavanja tlaka od prethodne rjeđe mreže. Zanimljivo je primijetiti da je broj linearnih iteracija znatno smanjen u odnosu na slučaj iste mreže s postavkom minimalnog broja jednadžbi na gruboj razini 5, slika 6.6, i veoma sličan broju iteracija u slučaju rjeđe mreže, slika 6.4.



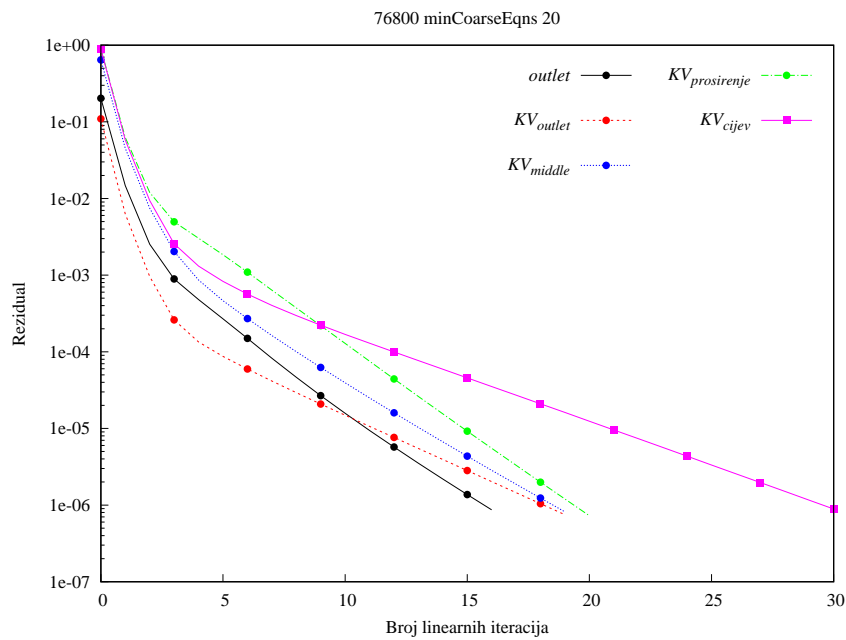
Slika 6.8: Konvergencija reziduala linearnog rješavača - mreža od 19200 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 100

Promatrajući sliku 6.8 može se uočiti daljnje smanjenje broja linearnih iteracija koji je u pojedinim slučajevima jednak ili čak manji od iste postavke minimalnog broja jednažbi na gruboj razini za rjeđu mrežu, slika 6.5. Za razliku od navedenog slučaja rjeđe mreže, ovdje je vidljiva razlika u konvergenciji između pojedinih slučajeva ovisno o mjestu zadavanja vrijednosti tlaka gdje slučaj KV cijev i KV middle konvergiraju brže od ostala 3 slučaja., ali se razlika smanjila u usporedbi sa slučajevima manjeg broja jednažbi na gruboj razini, slike 6.7 i 6.6.



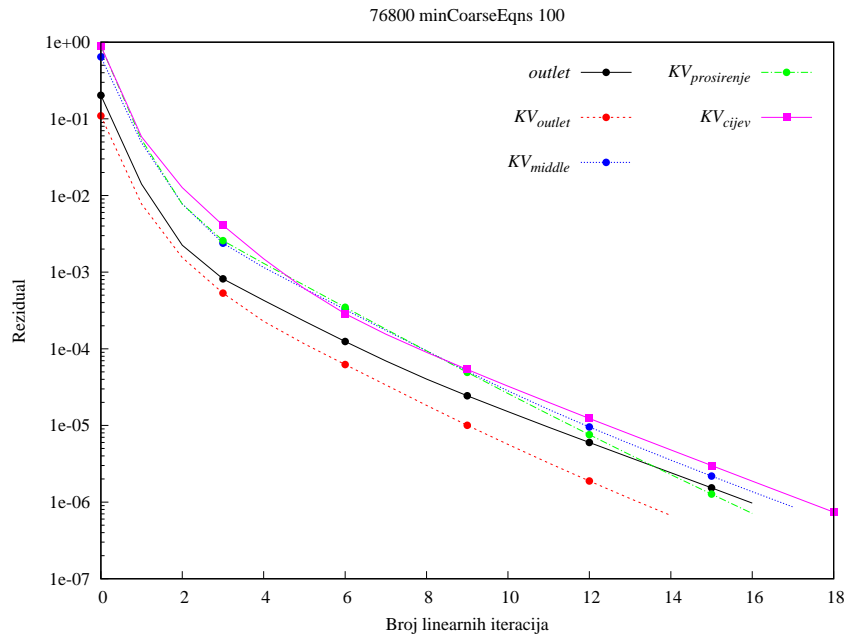
Slika 6.9: Konvergencija reziduala linearnog rješavača - mreža od 76800 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 5

Rezultati prikazani na slici 6.9 pripadaju najgušćoj izrađenoj mreži sa 76800 kontrolnih volumena i postavci minimalnog broja jednažbi na gruboj razini 5. Za razliku od prethodne dvije mreže, gdje se konvergencija u prvih par iteracija ne razlikuje, ovdje se početne vrijednosti reziduala vidljivo razlikuju za svaku varijantu zadavanja vrijednosti tlaka. Nadalje, uočava se značajna razlika u brzini konvergencije reziduala za svaku varijantu gdje broj linearnih iteracija varira od 23 do čak 43.



Slika 6.10: Konvergencija reziduala linearnog rješavača - mreža od 76800 kontrolnih volumena i minimalnim brojem jednažbi na gruboj razini 20

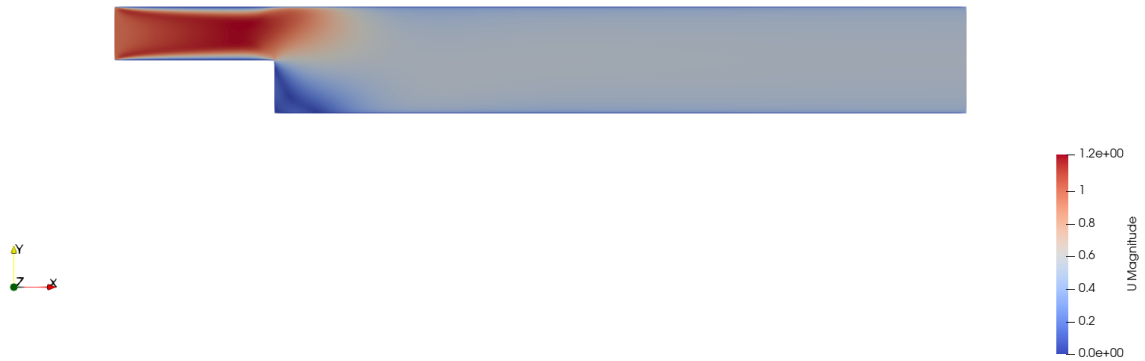
Na slici 6.10 također se uočava razlika u početnoj vrijednosti reziduala kao i na slici 6.9. U ovom slučaju se ponovo uočava drastično smanjenje broja linearnih iteracija s povećanjem minimalnog broja jednažbi na gruboj razini kao kod prethodne dvije mreže s istim postavkama, ali u nešto manjoj mjeri.



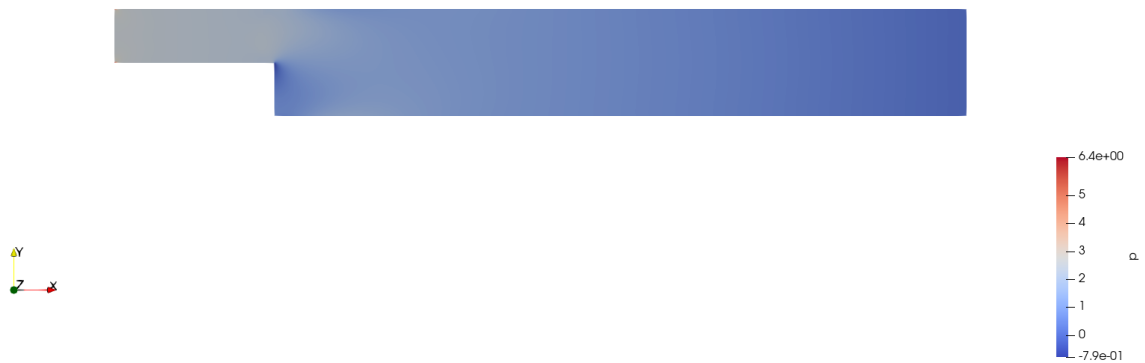
Slika 6.11: Konvergencija reziduala linearnog rješavača - mreža od 76800 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 100

Posljednji graf konvergencije reziduala prikazuje brzinu konvergencije s minimalnim brojem jednadžbi na gruboj razini 100. Kao i u prethodna dva slučaja na mreži sa 76800 kontrolnih volumena vidljiva je razlika u početnoj vrijednosti reziduala za različite varijante zadavanja tlaka. Potom se uočava smanjenje razlike u brzini konvergencije pojedinih slučajeva u odnosu na rezultate na slici 6.10. Ponovo se može uočiti da je broj linearnih iteracija za većinu varijanti zadavanja tlaka manji od prethodnog slučaja, slika 6.10, ali se smanjio manje u odnosu na iste postavke za prethodne dvije mreže.

Na slikama 6.12 i 6.13 prikazana su rješenja polja brzine i polja tlaka u 24000. nelinearnoj iteraciji na mreži sa 76800 kontrolnih volumena, minimalnim brojem jednadžbi na gruboj razini 20 te primjenom Gauss-Seidel algoritma kao zaglađivača. Rješenja polja brzine i tlaka bit će prikazana samo jednom s obzirom da njihov izgled nije predmet ovog rada.



Slika 6.12: Polje brzine na mreži od 76800 kontrolnih volumena u 24000. iteraciji



Slika 6.13: Polje tlaka na mreži od 76800 kontrolnih volumena u 24000. iteraciji

U 24000. iteraciji, na slikama 6.12 i 6.13, rješenje i dalje nije konvergiralo, ali se vidi formirani profil brzine na izlazu iz cijevi većeg promjera (slika 6.12) dok polje tlaka i dalje nije formirano (slika 6.13).

Kada se isti poredak jednadžbi, *strongest first*, primijeni koristeći kao zaglađivač ILU algoritam, rezultati su vidljivi u tablici 6.4. Ovdje je za usporedbu prikazan slučaj s primjenom Gauss-Seidel algoritma u 1000. nelinearnoj iteraciji. Oba slučaja provedena su na mreži sa 76800 kontrolnih volumena s postavkom minimalnog broja jednadžbi na gruboj razini 20.

min coarse equations		20	
pressure boundary condition		outlet	KV outlet
Gauss-Seidel	initial residual	3.63E-03	8.10E-04
	final residual	6.97E-07	7.16E-07
	number of coarse levels	8	7
	number of AMG iterations	10	8
	cumulative continuity error	1.13E+00	9.62E-11
ILU	initial residual	rješenje konvergira u 841. iteraciji	rješenje divergira u 176. iteraciji
	final residual		
	number of coarse levels		
	number of AMG iterations		
	cumulative continuity error		

Tablica 6.4: Tablica usporedbe rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU i Gauss-Seidel algoritma s minimalnim brojem jednadžbi na gruboj razini 20

U tablici 6.4 prikazano je da se u slučaju korištenja ILU algoritma kao zaglađivača rješenje divergira za slučaj KV outlet prije 1000. iteracije. Do divergencije dolazi zbog vrijednosti reziduala AMG rješavača (*tolerance*, slika 4.1) - umjesto da se rezidual smanjuje i konvergira k nekoj vrijednosti, njegova vrijednost se povećava sve dok se zbog prevelikog iznosa reziduala algoritam ne zaustavi. U slučaju outlet, rješenje je za ILU algoritam konvergiralo prije 1000. iteracije dok s Gauss-Seidelom kao zaglađivačem simulacija u 1000. iteraciji još uvijek traje.

Zanimljivo je primijetiti da su različiti rezultati dobiveni bez primjene postavke *renumberMesh* prije provedbe simulacija koja mijenja brojčane nazive kontrolnih volumena i time sam poredak jednadžbi u finoj matrici tako što približava vandijagonalne elemente dijagonali, odnosno grupira ih u vrpce oko dijagonale. Rezultati prije naredbe *renumberMesh* mogu se vidjeti u tablici 6.5:

min coarse equations		20	
pressure boundary condition		outlet	KV outlet
Gauss-Seidel	initial residual	4.02E-03	5.10E-03
	final residual	9.34E-07	6.62E-07
	number of coarse levels	7	7
	number of AMG iterations	10	11
	cumulative continuity error	1.72E+00	-1.84E-11
ILU	initial residual	rješenje divergira u 763. iteraciji	rješenje divergira u 203. iteraciji
	final residual		
	number of coarse levels		
	number of AMG iterations		
	cumulative continuity error		

Tablica 6.5: Tablica rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU algoritma s minimalnim brojem jednadžbi na gruboj razini 20

Iz tablice 6.5 vidljivo je da se primjenom ILU algoritma na *strongest first* poretku bez promjene poretka jednadžbi u finoj matrici rješenje divergira prije 1000. iteracije za obje varijante zadavanja tlaka. Promatrajući rezultate dobivene korištenjem Gauss-Seidel algoritma kao zaglađivača, vidljivo je da promjene u odnosu na tablicu 6.7 nisu od velikog značaja kao za ILU algoritam.

Rezultati na mreži s izduženim kontrolnim volumenima

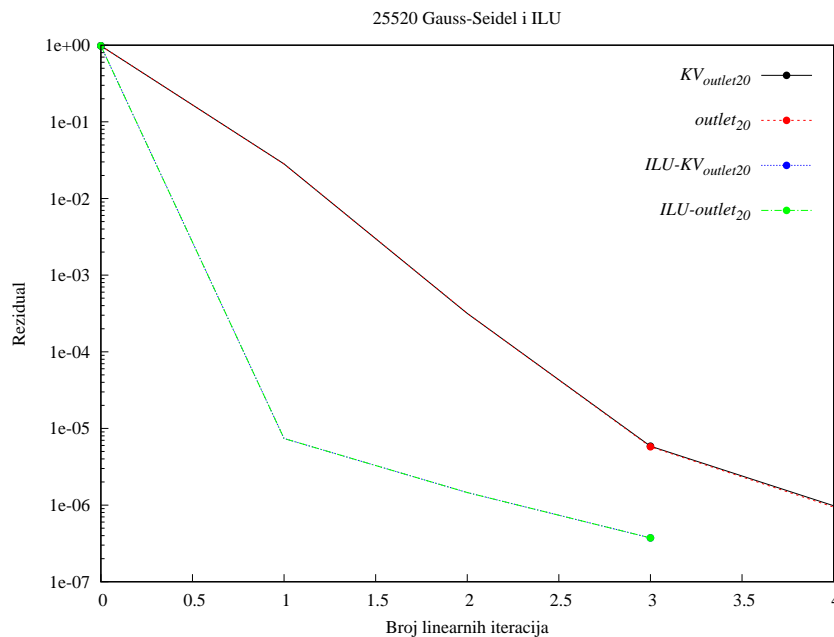
Poredak *strongest first* testiran je na mreži s 25520 kontrolnih volumena čiji su volumeni izduženi u smjeru osi x kako je opisano u poglavlju 5.. Simulacije su provedene s Gauss-Seidel i ILU algoritmima kao zaglađivačima, od kojih je za svaki zaglađivač provedena simulacija za dvije varijante zadavanja tlaka - outlet i KV outlet, te su dobiveni rezultati u 1000. iteraciji prikazani u tablici:

min coarse equations		20	
pressure boundary condition		outlet	KV outlet
Gauss-Seidel	initial residual	9.82E-01	9.81E-01
	final residual	9.42E-07	9.82E-07
	number of coarse levels	7	7
	number of AMG iterations	4	4
	cumulative continuity error	-9.15E-06	2.16E-12
ILU	initial residual	9.82E-01	9.81E-01
	final residual	3.73E-07	3.40E-07
	number of coarse levels	7	7
	number of AMG iterations	3	3
	cumulative continuity error	-8.63E-05	2.27E-12

Tablica 6.6: Tablica rezultata simulacija na mreži s 25520 kontrolnih volumena korištenjem Gauss-Seidel i ILU algoritma s minimalnim brojem jednadžbi na gruboj razini 20

Iz rezultata u tablici 6.6 može se uočiti da s istim postavkama kao u slučaju prikazanom na tablici 6.4, ali primijenjenim na mreži s izduženim kontrolnim volumenima slučajevi s ILU algoritmom kao zaglađivačem i dalje konvergiraju k rješenju u 1000. iteraciji. Nadalje, može se uočiti kako je broj iteracija AMG rješavača u slučaju korištenja ILU algoritma manji od slučaja s Gauss-Seidel algoritmom.

Kako bi se bolje prikazala razlika između dva algoritma zaglađivača, konvergencija reziduala za sva 4 slučaja u 1000. nelinearnoj iteraciji prikazana je na slici 6.14:



Slika 6.14: Konvergencija reziduala linearnog rješavača - mreža od 25520 kontrolnih volumena i minimalnim brojem jednadžbi na gruboj razini 20

Iz slike 6.14 vidi se ono što je brojčano prikazano u tablici, a to je veća brzina konvergencije reziduala za obje varijante zadavanja tlaka outlet i KV outlet pri korištenju ILU algoritma kao zaglađivača te kako je konvergencija pri korištenju ILU algoritma mnogo strmija u odnosu na konvergenciju dobivenu Gauss-Seidel algoritmom. Valja naglasiti kako je poredak jednadžbi na gruboj razini za prikazane rezultate i dalje *strongest first*.

6.3.2. *fine order* poredak

Mijenjajući poredak jednadžbi na gruboj razini utječe se na rad zaglađivača i konvergenciju rješenja. Gauss-Seidel i ILU algoritmi su već isprobani na poretku jednadžbi *strongest first* stoga se za usporedbu u ovom potpoglavlju prikazuju rezultati dobiveni u 1000. iteraciji primjenom prvo jednog pa drugog algoritma na najgušćoj mreži (od 76800 kontrolnih volumena), promjenom minimalnog broja jednadžbi na gruboj razini kao i u prethodnim rezultatima, ali samo za dva slučaja zadavanja tlaka - outlet i KV outlet. Rezultati su prikazani u sljedećoj tablici:

min coarse equations		5		20		100	
pressure boundary condition		outlet	KV outlet	outlet	KV outlet	outlet	KV outlet
Gauss-Seidel	initial residual	3.55E-03	7.60E-04	3.59E-03	7.54E-04	3.51E-03	7.51E-04
	final residual	7.88E-07	8.16E-07	6.08E-07	6.75E-07	6.25E-07	9.28E-07
	number of coarse levels	9	8	7	7	6	6
	number of AMG iterations	12	9	9	8	10	6
	cumulative continuity error	3.27E+00	8.16E-07	1.06E+00	-1.75E-12	2.72E-01	-4.02E-11

ILU	initial residual	rješenje divergira	rješenje divergira	rješenje divergira	rješenje divergira	3.51E-03	rješenje divergira
	final residual					5.54E-07	
	number of coarse levels					6	
	number of AMG iterations					7	
	cumulative continuity error					1.74E+00	

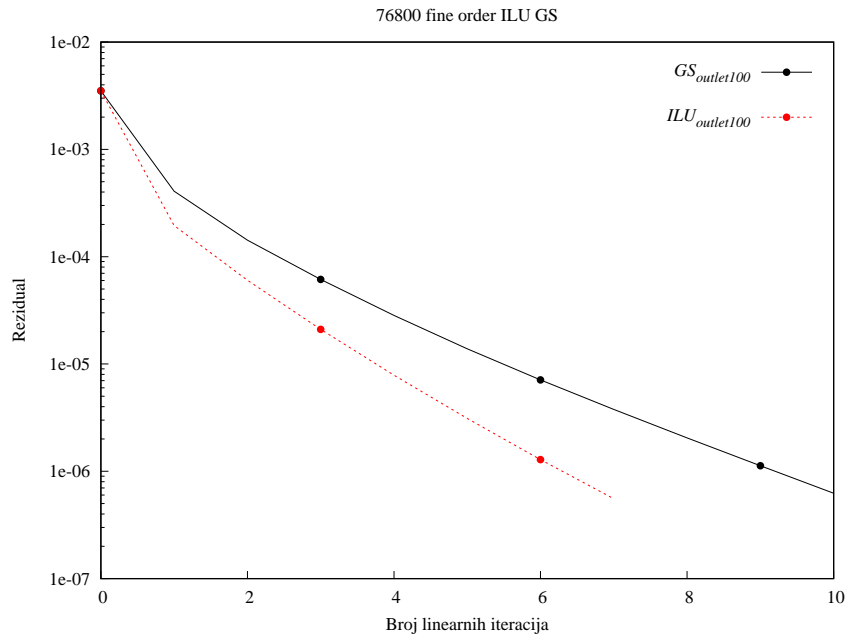
Tablica 6.7: Tablica rezultata simulacija u 1000. iteraciji na mreži sa 76800 kontrolnih volumena korištenjem Gauss-Seidel i ILU algoritma s poretkom jednadžbi kao u finoj matrici

Iz tablice 6.7 se može uočiti da primjenom Gauss-Seidel algoritma kao zaglađivača na promijenjenom poretku jednadžbi na gruboj razini algoritam i dalje funkcionira bez prekida za sve tri postavke minimalnog broja jednadžbi na gruboj razini. Vidljivo je smanjenje broja grubih razina te smanjenje broja linearnih iteracija s povećanjem broja jednadžbi na gruboj razini kao što su pokazali slučajevi sa *strongest first* poretkom. Kumulativna greška kontinuiteta je također veća za zadavanje vrijednosti tlaka na cijeloj izlaznoj plohi (outlet) nego što je za zadavanje tlaka u samo jednom kontrolnom volumenu (KV outlet).

Promatrajući rezultate dobivene korištenjem ILU algoritma kao zaglađivača, uočava se da za broj jednadžbi na gruboj razini 5 i 20 u oba slučaja zadavanja tlaka rješenje divergira, isto kao i za broj jednadžbi na gruboj razini 100 u slučaju KV outlet. Samo slučaj outlet za minimalan broj jednadžbi na gruboj razini 100 i dalje konvergira k rješenju u 1000. iteraciji. Korištenje ILU algoritma pokazuje manji broj linearnih iteracija AMG rješavača u usporedbi s Gauss-Seidelom. Međutim, vidljivo je da se postavljanjem premalog minimalnog broja jednadžbi na gruboj razini u kombinaciji s lokacijom zadavanja vrijednosti tlaka simulacija s ILU algoritmom prekida (rješenje divergira), što korištenjem Gauss-Seidela nije slučaj, ali isto tako da se postavljanjem određenog broja jednadžbi na gruboj razini može ubrzati konvergencija.

Izrade li se grafovi koji prikazuju brzinu konvergencije za varijantu outlet s minimalnim brojem jednadžbi na gruboj razini 100 za oba korištena algoritma (pošto su to

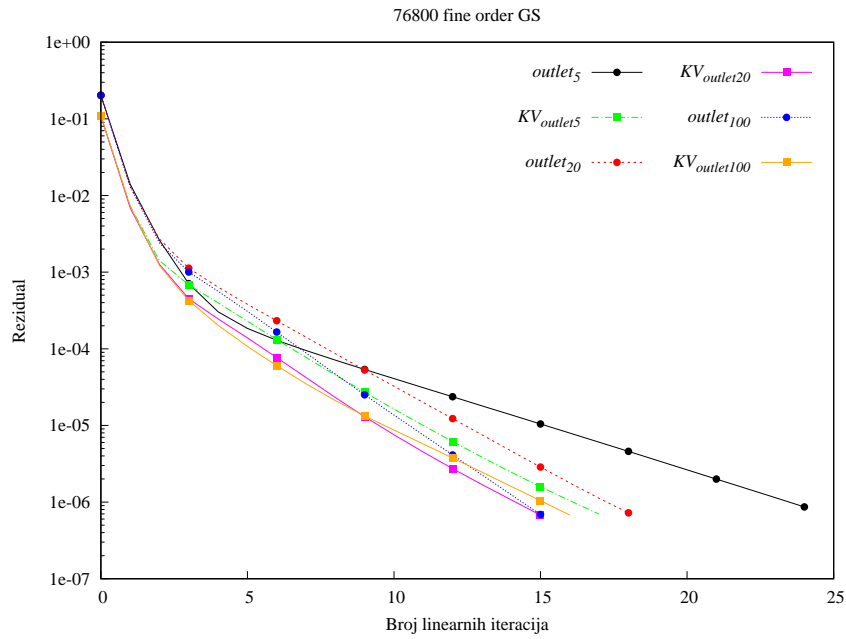
varijante u kojima za oba algoritma konvergencija još traje u 1000. iteraciji), dobivaju se rezultati na slici 6.15:



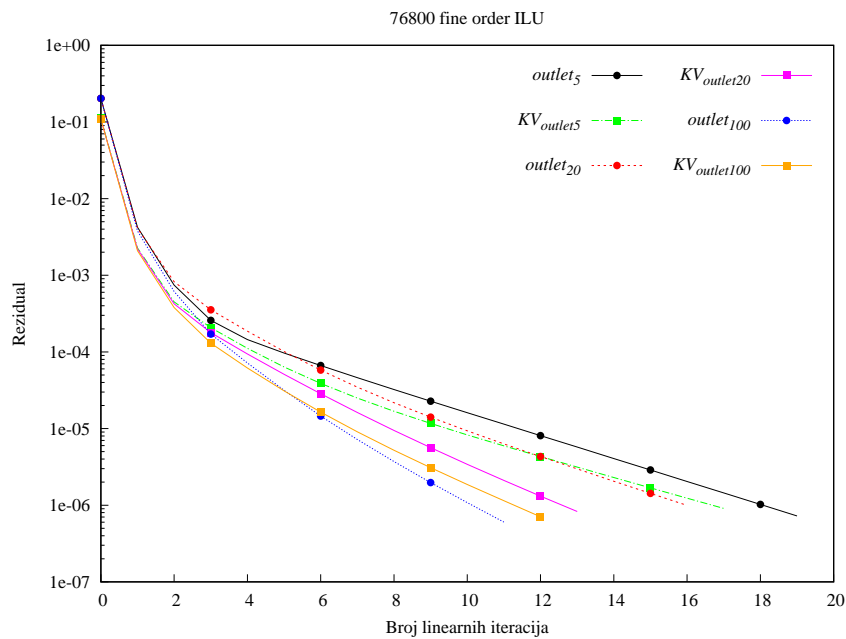
Slika 6.15: Konvergencija reziduala linearnog rješavača s primjenom Gauss-Seidel i ILU algoritma na *fine order* poretku

Promatrajući konvergenciju reziduala na slici 6.15 može se uočiti brža konvergencija reziduala korištenjem ILU algoritma kao zaglađivača u odnosu na Gauss-Seidel za iste postavke rješavača i poredak jednadžbi na gruboj razini *fine order*.

Kako bi se bolje proučila konvergencija reziduala za dva različita algoritma koji su korišteni kao zaglađivači, izrađeni su dijagrami konvergencije reziduala za sve ispitane slučajeve lokacije zadavanja vrijednosti tlaka i sve tri postavke minimalnog broja jednadžbi na gruboj razini, posebno za svaki algoritam. Rezultati su zbog divergencije rješenja kod ILU algoritma uzeti u 10. nelinearnoj iteraciji gdje sve simulacije još traju. Rezultati dobiveni korištenjem Gauss-Seidel algoritma vidljivi su na slici 6.16, a oni dobiveni korištenjem ILU algoritma na slici 6.17.



Slika 6.16: Konvergencija reziduala linearnog rješavača s primjenom Gauss-Seidel algoritma na *fine order* poretku



Slika 6.17: Konvergencija reziduala linearnog rješavača s primjenom ILU algoritma na *fine order* poretku

Uspoređujući rezultate na slikama 6.16 i 6.17 može se odmah uočiti manji raspon broja linearnih iteracija korištenjem ILU algoritma u odnosu na Gauss-Seidel. Nadalje, na obje slike može se uočiti da slučajevi zadavanja tlaka KV outlet za postavke minimalnog broja jednadžbi 5 i 20 konvergiraju brže od svojih parova outlet za iste postavke, dok je u slučaju minimalnog broja jednadžbi 100 slučaj obrnut - varijanta outlet konvergira brže od varijante KV outlet, ali samo za 1 linearnu iteraciju više. Na slici 6.15 uočava se pravilnost u smanjivanju broja iteracija linearnog rješavača za oba slučaja zadavanja tlaka s povećanjem minimalnog broja jednadžbi na gruboj razini korištenjem ILU algoritma kao zaglađivača. Korištenjem Gauss-Seidel algoritma ova pravilnost nije uočena, slika 6.16.

Rezultati na tetraedarskim mrežama

Rad ILU algoritma isproban je na još dvije mreže objašnjene u poglavlju 5. koje umjesto heksaedara kao kontrolne volumene imaju tetraedre, prikazane na slikama 5.6 i 5.7. Poredak jednadžbi je i dalje *fine order* te su simulacije provedene s varijantama zadavanja tlaka outlet i KV outlet za minimalan broj jednadžbi na gruboj razini 5, 20 i 100. Usporedba rezultata dobivenih na dvije tetraedarske mreže u 10. nelinearnoj iteraciji prikazana je u tablici 6.8.

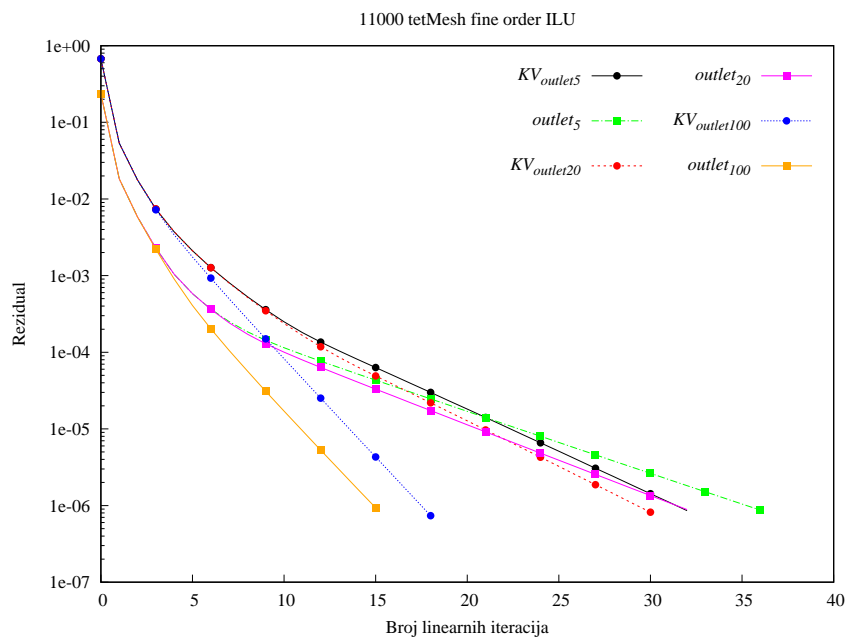
min coarse equations		5		20		100	
pressure boundary condition		outlet	KV outlet	outlet	KV outlet	outlet	KV outlet
11046 cells	initial residual	2.37E-01	6.75E-01	2.37E-01	6.75E-01	2.37E-01	6.75E-01
	final residual	8.70E-07	8.57E-07	8.85E-07	8.19E-07	9.37E-07	7.38E-07
	number of coarse levels	7	7	6	6	4	4
	number of AMG iterations	36	32	32	30	15	18
	cumulative continuity error	-5.14E-04	1.13E-14	-4.32E-04	1.10E-15	-4.54E-04	1.43E-14
6228 cells	initial residual	1.26E-01	7.20E-01	1.26E-01	7.20E-01	1.26E-01	7.20E-01
	final residual	8.66E-07	8.44E-07	8.96E-07	7.42E-07	6.62E-07	9.39E-07
	number of coarse levels	6	7	5	5	4	4
	number of AMG iterations	30	35	18	26	15	17
	cumulative continuity error	1.33E-03	7.65E-15	9.40E-04	4.10E-15	1.02E-03	-1.01E-14

Tablica 6.8: Tablica rezultata simulacija u 10. iteraciji na tetraedarskim mrežama korištenjem ILU algoritma s poretkom jednadžbi kao u finoj matrici

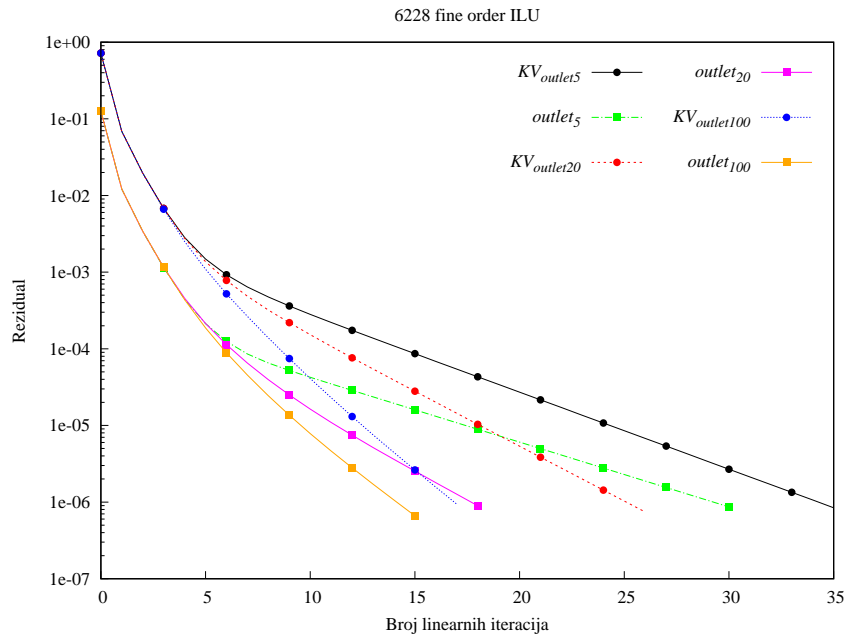
Iz rezultata u tablici 6.8 može se uočiti kako se primijecena pojava veće kumulativne greške kontinuiteta za slučaj outlet u donosu na ostale varijante zadavanja tlaka ponavlja i kod tetraedarskih mreža. Broj kreiranih grubih razina je sličan za obje mreže, dok

je kod nekih varijanti zadavanja tlaka manji za mrežu sa 6228 kontrolnih volumena. Broj iteracija linearnog rješavača ne pokazuje nikakav uzorak s obzirom na varijantu zadavanja tlaka, ali se smanjuje s povećanjem broja jednažbi na gruboj razini za obje tetraedarske mreže.

Kako bi se bolje uočile potencijalne razlike u rezultatima između dvije mreže, izrađeni su dijagrami konvergencije reziduala prikazani na slikama 6.18 i 6.19.



Slika 6.18: Konvergencija reziduala linearnog rješavača s primjenom ILU algoritma na tetraedarskoj mreži



Slika 6.19: Konvergencija reziduala linearnog rješavača s primjenom ILU algoritma na tetraedarskoj mreži s većim kontrolnim volumenima na izlazu

Promatrajući slike 6.18 i 6.19 može se učiti da se brzina konvergencije ne razlikuje značajno bez obzira na manji broj i povećanje kontrolnih volumena na izlazu kod mreže sa 6228 kontrolnih volumena. Na obje slike najbržu konvergenciju pokazuju slučajevi s minimalni brojem jednadžbi na gruboj razini 100, a najsporiju konvergenciju slučajevi s minimalnim brojem jednadžbi 5.

Zbog različitog oblika kontrolnih volumena, zanimljivo je usporediti rezultate dobivene korištenjem ILU algoritma kao zaglađivača s *fine order* poretkom jednadžbi za tetraedarsku i heksaedarsku mrežu. Rezultati u 10. nelinearnoj iteraciji za dvije različite mreže prikazani su u tablici 6.9 gdje je oznakom '76800 hexMesh' označena heksaedarska mreža sa 76800 kontrolnih volumena, a oznakom '11046 tetMesh' tetraedarska mreža s 11046 kontrolnih volumena.

min coarse equations		5		20		100	
pressure boundary condition		outlet	KV outlet	outlet	KV outlet	outlet	KV outlet
78600 hexMesh	initial residual	2.03E-01	1.11E-01	2.03E-01	1.10E-01	2.03E-01	1.09E-01
	final residual	7.25E-07	9.08E-07	9.93E-07	8.27E-07	6.02E-07	7.07E-07
	number of coarse levels	8	9	7	7	6	6
	number of AMG iterations	19	17	16	13	11	12
	cumulative continuity error	-1.02E-01	-3.34E-12	-5.70E-03	-2.69E-12	-6.42E-04	-2.46E-12
11046 tetMesh	initial residual	2.37E-01	6.75E-01	2.37E-01	6.75E-01	2.37E-01	6.75E-01
	final residual	8.70E-07	8.57E-07	8.85E-07	8.19E-07	9.37E-07	7.38E-07
	number of coarse levels	7	7	6	6	4	4
	number of AMG iterations	36	32	32	30	15	18
	cumulative continuity error	-5.14E-04	1.13E-14	-4.32E-04	1.10E-15	-4.54E-04	1.43E-14

Tablica 6.9: Tablica rezultata simulacija u 10. iteraciji na tetraedarskim mrežama korištenjem ILU algoritma s poretkom jednadžbi kao u finoj matrici

U tablici 6.9 može se uočiti da su vrijednosti početnih i krajnjih reziduala veoma slične za obje mreže u 10. nelinearnoj iteraciji. Kumulativna greška kontinuiteta je nekoliko redova veličine manja kod tetraedarske mreže u odnosu na heksaedarsku dok se ponovo može uočiti manja greška za varijante outlet kod obje mreže. Primjećuje se da je broj kreiranih grubih razina manji kod tetraedarske mreže u odnosu na heksaedarsku. Zanimljivo je uočiti da je broj iteracija linearnog rješavača znatno veća i kod tetraedarske mreže u odnosu na heksaedarsku mrežu u sva tri slučaja minimalnog broja jednadžbi na gruboj razini i obje varijante zadavanja tlaka.

6.4. Snaga veza između jednadžbi

Kako je objašnjeno u poglavlju 4., snaga veze između jednadžbi utječe na odabir jednadžbi iz fine matrice koje nakon *coarsening*-a ostaju na gruboj razini. Dosadašnji rezultati izrađeni su sa snagom veze $\varepsilon = 0.2$. Provedene su simulacije na najgušćoj mreži, 76800 kontrolnih volumena, koristeći ILU i Gauss-Seidel algoritam kao zaglađivač i poredak jednadžbi kao u finoj matrici. Tako je moguća usporedba s rezultatima iz tablice 6.7. Snaga veze promijenjena je u izvornom kodu i za sljedeće rezultate ona iznosi $\varepsilon = 0.3$.

min coarse equations		5		20		100	
pressure boundary condition		outlet	KV outlet	outlet	KV outlet	outlet	KV outlet
Gauss-Seidel	initial residual	3.55E-03	7.60E-04	3.59E-03	7.54E-04	3.51E-03	7.51E-04
	final residual	7.88E-07	8.16E-07	6.08E-07	6.75E-07	6.25E-07	9.28E-07
	number of coarse levels	9	8	7	7	6	6
	number of AMG iterations	12	9	9	8	10	6
	cumulative continuity error	3.27E+00	8.16E-07	1.06E+00	-1.75E-12	2.72E-01	-4.02E-11

ILU	initial residual	rješenje divergira	rješenje divergira	rješenje divergira	rješenje divergira	3.51E-03	rješenje divergira
	final residual					5.54E-07	
	number of coarse levels					6	
	number of AMG iterations					7	
	cumulative continuity error					1.74E+00	

Tablica 6.10: Tablica rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU i Gauss-Seidel algoritma s poretkom jednadžbi kao u finoj matrici i snagom veze 0.3

Usporede li se rezultati iz tablice 6.10 s rezultatima iz tablice 6.7, može se uočiti da su u potpunosti jednaki. Promjena snage veze na istim postavkama AMG rješavača i istim poretkom jednadžbi, *fine order*, nije pokazala nikakve promjene u rješenjima dobivenih u 1000. iteraciji bez obzira na to je li korišten Gauss-Seidel ili ILU algoritam kao zaglađivač.

6.5. Promjena sume elemenata u retku matrice prolongacije

Matrica prolongacije prebacuje grešku s grube razine na finu razinu pomoću interpolacijskih faktora $w_{i,j}$, poglavlje 3.4.2.. Uobičajeno je da ako je na finoj razini matrica dijagonalno dominantna (veličina elementa na dijagonali je veća od sume veličina vandiagonalnih elemenata), onda je $\sum_i w_{i,j} > 1$ u odgovarajućem retku matrice prolongacije \mathbf{P} . U ovom slučaju unutar izvornog koda promijenjena suma elemenata u retku matrice prolongacije, odnosno nametnuta je sumacija svih elemenata retka matrice u 1, $\sum_i w_{i,j} = 1$. Simulacija je provedena na mreži od 76800 kontrolnih volumena koristeći ILU algoritam kao zaglađivač s poretkom jednadžbi na gruboj razini istim kao na finoj razini (*fine order*). Variran je minimalan broj jednadžbi na gruboj razini, kao u dosadašnjim slučajevima, i mijenjana je lokacija zadavanja vrijednosti tlaka. Dobiveni rezultati vidljivi su u sljedećoj tablici.

min coarse equations		5	
pressure boundary condition		outlet	KV outlet
ILU	initial residual	rješenje divergira u 287. iteraciji	rješenje divergira u 6. iteraciji
	final residual		
	number of coarse levels		
	number of AMG iterations		
	cumulative continuity error		

min coarse equations		20	
pressure boundary condition		outlet	KV outlet
ILU	initial residual	3.54E-03	rješenje divergira u 89. iteraciji
	final residual	5.97E-07	
	number of coarse levels	7	
	number of AMG iterations	8	
	cumulative continuity error	1.20E+01	

min coarse equations		100	
pressure boundary condition		outlet	KV outlet
ILU	initial residual	rješenje divergira u 970. iteraciji	rješenje divergira u 945. iteraciji
	final residual		
	number of coarse levels		
	number of AMG iterations		
	cumulative continuity error		

Tablica 6.11: Tablica rezultata simulacija na mreži sa 76800 kontrolnih volumena korištenjem ILU algoritma s poretkom jednadžbi kao u finoj matrici i promjenom sume elemenata u retku matrice prolongacije

Iz tablice 6.11 mogu se uočiti divergencije rješenja kod pojedinih slučajeva za svaku od postavki minimalnog broja jednadžbi 5, 20 i 100. Zanimljivo je uočiti da se s promjenom sume elemenata u retku matrice prolongacije simulacija s istim ostalim postavkama prekine prije 1000. iteracije što nije bio slučaj prije promjene sume elemenata u retku gdje je simulacija u 1000. iteraciji i dalje konvergirala k rješenju (primjer minimalnog broja jednadžbi na gruboj razini 100, varijanta outlet, tablica 6.7. Također, simulacija koja se prije promjene sume elemenata u retku matrice prolongacije prekidala prije 1000 iteracije zbog divergencije rješenja (*minCoarseEquations* 20, outlet u tablici 6.7), s promjenom sume elemenata u retku u 1000. iteraciji i dalje konvergira k rješenju.

Promjene u dobivenim rezultatima se mogu još bolje uočiti prikazujući broj dijagonalno dominantnih jednadžbi na grubim razinama koji se mijenja promjenom postavke sume elemenata u retku matrice prolongacije. U sljedećoj tablici prikazana su dva slučaja s istim postavkama AMG rješavača, *fine order* poretkom jednadžbi na gruboj razini i tlakom zadanim na lokaciji outlet. Jedina razlika je promjena sume elemenata

u retku matrice prolongacije.

min coarse equations		100	
pressure boundary condition		outlet	
suma redaka konzistentna			
Razina	Broj jednadžbi	Dijagonalno dominantne jednadžbe	Dijagonalno slabe jednadžbe
1	76800	160	0
2	38400	160	0
3	13122	121	107
4	3763	74	138
5	958	33	64
6	253	16	15

suma redaka u 1			
Razina	Broj jednadžbi	Dijagonalno dominantne jednadžbe	Dijagonalno slabe jednadžbe
1	76800	160	0
2	38400	80	0
3	13159	41	86
4	3770	26	38
94	971	16	55
6	244	9	6

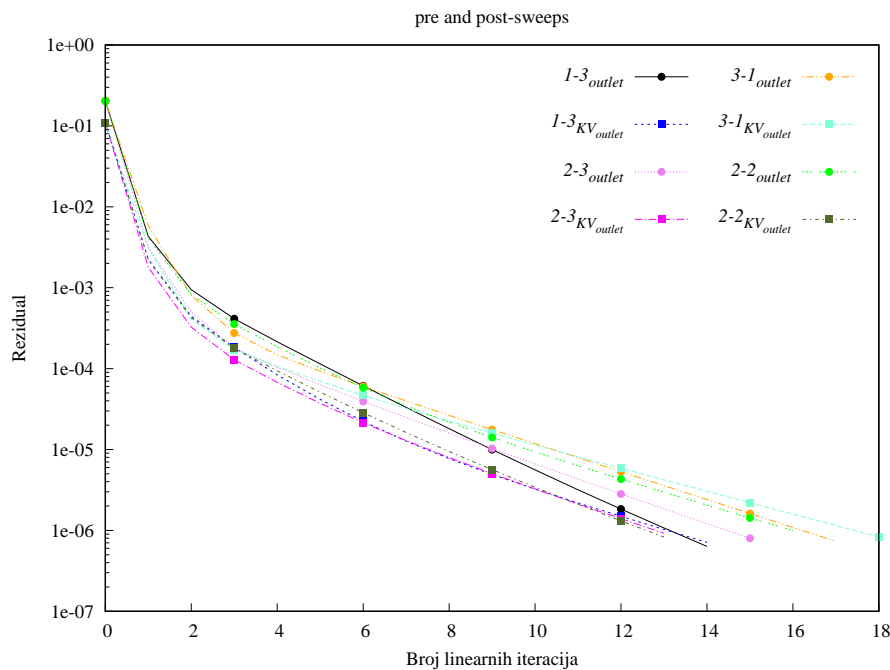
Tablica 6.12: Usporedba rezultata simulacija prije i nakon promjene sume elemenata u retku matrice prolongacije

Iz tablice 6.12 može se uočiti da se broj dijagonalno dominantnih jednadžbi za sumu redaka matrice prolongacije nametnutu u 1 smanjuje značajno više sa svakom grubom razinom u odnosu na broj dijagonalno dominantnih jednadžbi za konzistentnu sumu redaka. Na pojedinim razinama broj dijagonalno dominantnih jednadžbi za sumu redaka u 1 je dva do tri puta manji u odnosu na istu razinu za konzistentnu sumu redaka.

6.6. Broj primjene zaglađivača prije i nakon AMG rješavača

Posljednja postavka koja je mijenjana u ovom radu jest broj primjene algoritma zaglađivača prije i nakon AMG rješavača. Algoritmi koji se koriste kao zaglađivači imaju ulogu eliminiranja visokofrekventnih komponenti greški prije AMG ciklusa te zaglađivanja rješenja nakon AMG ciklusa, točnije eliminiranja komponenti greški koje je AMG rješavač pretvorio u visokofrekventne. Njihovom primjenom dobiva se brža konvergencija rješenja i potrebno je naći odgovarajući broj primjene algoritma zaglađivača

prije i nakon AMG ciklusa koji će dati najbržu konvergenciju rješenja. Literatura nalaže da bi broj primjene zaglađivača prije i nakon AMG ciklusa trebao biti jednak, odnosno da postoji simetrija čitavog ciklusa kako bi se očuvale svojstvene vrijednosti matrice linearnog sustava (objašnjeno u poglavlju 3.), na primjer: $\mathbf{S}^2\mathbf{MS}^2$. U ovom radu izmijenjeno je nekoliko opcija i provedene su simulacije na mreži od 76800 kontrolnih volumena, s poretkom na gruboj razini *fine order*, korištenjem ILU algoritma kao zaglađivača, snagom veze $\varepsilon = 0.2$ i konzistentne sume redaka matrice prolongacije. Za simulacije su odabrane varijante zadavanja vrijednosti tlaka outlet i KV outlet, uz minimalan broj jednadžbi na gruboj razini 20, čiji su rezultati u 10. nelinearnoj iteraciji vidljivi na sljedećoj slici:



Slika 6.20: Konvergencija reziduala linearnog rješavača s različitim brojem primjene ILU algoritma na *fine order* poretku

Na dijagramu na slici 6.20 prikazani su različiti slučajevi broja primjene ILU algoritma označeni s dva broja, na primjer 1-3. Prvi broj označava broj primjene zaglađivača prije AMG ciklusa (eng. *pre-sweep*), a drugi broj primjene zaglađivača nakon AMG ciklusa (eng. *post-sweep*). Ukupno je prikazano 8 slučajeva, za svaku kombinaciju *pre* i *post-sweep* varijanta zadavanja tlaka outlet i KV outlet.

Iz dijagrama je vidljivo da je konvergencija brža za varijanta KV outlet (prikazani kva-

dratićima) no što je za varijante outlet (prikazani kružićima), bez obzira broj primjene zaglađivača. Može se uočiti kako su slučajevi simetrične primjene zaglađivača, 2-2, otprilike u sredini po brzini konvergencije za obje varijante zadavanja tlaka. Kombinacija primjene zaglađivača 3-1 pokazala je najsporiju konvergenciju od svih slučajeva outlet (narančasta linija) i svih slučajeva KV outlet (svjetlo plava linija). Zanimljivo je primijetiti kako je kombinacija primjene zaglađivača 2-3 pokazala najbržu brzu konvergenciju od slučajeva outlet u početku (svjetlo ružičasta linija s kružićima) dok ju u 10. iteraciji prestiže slučaj 1-3, i podjednako brzu konvergenciju kao kombinacija 2-2 od slučajeva KV outlet (tamnije ružičasta linija s kvadratićima). Posljednja kombinacija, 1-3, pokazala je najbržu konvergenciju za slučaj outlet, ali zato srednje brzu konvergenciju za slučaj KV outlet.

6.7. Zaključak

U ovom poglavlju prikazani su rezultati simulacija provedenih na različitim proračunskim mrežama s različitim postavkama AMG rješavača i algoritama zaglađivača kako bi se ustanovilo kako pojedine postavke rješavača utječu na konvergenciju rješenja te koje postavke rješavača najbolje odgovaraju različitim algoritmima koji se koriste kao zaglađivači rješenja. Rješenja su prikazana tablično, zamijećene promjene detaljno opisane te, tamo gdje je potrebno, izrađeni grafovi kako bi se promotrila razlika u brzini konvergencije reziduala za pojedine slučajeve. Uočene su promjene u dobivenim rješenjima ovisno o različitim postavkama algoritma i promjenama u izvornom kodu.

Sljedeće poglavlje namijenjeno je diskusiji o dobivenim rezultatima i donošenju rezultatima potkrijepljenih zaključaka o postavkama AMG rješavača i popratnih algoritama.

7 | Diskusija

7.1. Uvod

Ovo poglavlje posvećeno je diskusiji o rezultatima prikazanim i opisanim u prethodnom poglavlju. Utvrdit će se uzročno-posljedična veza između postavki multigrid algoritma i dobivenih rezultata te sažeti sve uspostavljene veze u tabličnom prikazu kako bi se budućim korisnicima AMG rješavača olakšao proces prilagođavanja postavki.

7.2. Diskusija o rezultatima

7.2.1. *strongest first* poredak jednadžbi

Iz navedenih rezultata može se zaključiti da *strongest first* poredak zaista odgovara zaglađivaču Gauss-Seidel zbog načina na koji on rješava linearni sustav sustavno propagirajući kroz matricu linearnog sustava. Ako jednadžbe s najviše utjecaja bivaju zadržane na gruboj razini, njihova korekcija putem matrice prolongacije utječe na korigiranje većeg broja jednadžbi utjecajnika čime se pospješuje širenje informacija. Gauss-Seidel potom propagira redom kroz matricu u kojoj su jednadžbe primile korekciju same od sebe ili od svojih utjecaja što doprinosi konvergenciji prilikom rješavanja linearnog sustava. Nadalje, testiranjem ILU algoritma s istim poretkom jednadžbi *strongest first* potvrđena je hipoteza da ILU algoritmu takav poredak jednadžbi ne odgovara, ali samo za određene postavke zadavanja vrijednosti tlaka. Rezultat testiranja ILU algoritma je prekid simulacije zbog divergencije reziduala AMG rješavača varijantu zadavanja tlaka u jednom kontrolnom volumenu (KV outlet) u kojoj je s Gauss-Seidel algoritmom kao

zaglađivačem i u 1000. iteraciji rezidual pokazao konvergenciju ka rješenju. Statički ILU algoritam, kako je navedeno u poglavlju 3.3.2., prilikom **LU** faktorizacije ispušta pojedine elemente iz matrice kako bi zadržao isti uzorak popunjenosti matrice **A** nakon faktorizacije. Zbog *strongest first* rasporeda jednadžbi na gruboj razini, svaka nova gruba razina ima drugačiji redosljed jednadžbi. Sa svakom novom razinom ILU algoritam radi nepotpunu faktorizaciju stvarajući novu približno točnu matricu s kojom izglađuje rješenje (korekciju) dobiveno na toj razini. Potom se stvaranjem nove razine ponovo mijenja broj i redosljed jednadžbi nad kojima ILU algoritam operira i korekcija koja je zaglađena ILU algoritmom na matrici s prethodnim poretkom se sad koristi za izračun nove korekcije nad drugačijim poretkom. Zato ILU algoritam unosi grešku koja se povećava sa svakom novom razinom. *Strongest first* poredak jednadžbi ne zadržava vrpčastost matrice koja ILU algoritmu odgovara već rasprši elemente dalje od dijagonale prilikom **ILU** faktorizacije. Na taj način ILU eliminira neke od jakih veza između jednadžbi čime se smanjuje (na mjestima i onemogućuje) širenje informacija i dolazi do zastoja algoritma. U varijanti zadavanja tlaka na čitavoj izlaznoj plohi, ILU algoritam pokazao je bržu konvergenciju u odnosu na Gauss-Seidel. Kod ove varijante zadavanja tlaka unosi se velika kumulativna greška kontinuiteta zbog nametnute uniformnosti tlaka na izlaznoj plohi. Kako je već navedeno, ILU algoritam ispušta pojedine elemente matrice tijekom svoje faktorizacije zbog čega je u ovom slučaju moguće, zbog velikog broja elemenata u kojima je zadana vrijednost tlaka, da nije ispušteno previše značajnih elemenata koji pomažu dostizanju rješenja, ali s jako velikom greškom kontinuiteta u odnosu na Gauss-Seidel algoritam. Navedena tvrdnja može se uočiti na tablici 7.1 gdje je vidljivo da je kumulativna greška kontinuiteta za Gauss-Seidel algoritam mnogo redova veličine manja od one za ILU algoritam (za iste postavke u nelinearnoj iteraciji u kojoj je simulacije s ILU algoritmom konvergirala).

Non-linear iteration		841
min coarse equations		20
pressure boundary condition		outlet
ILU	cumulative continuity error	-3.86E+22
Gauss-Seidel	cumulative continuity error	1.1507

Tablica 7.1: Tablica usporedbe kumulativne greške kontinuiteta na mreži od 76800 kontrolnih volumena s minimalnim brojem jednadžbi na gruboj razini 20

Nadalje, uočene su razlike u rješenjima dobivenim renumeriranjem kontrolnih volumena mreže u odnosu na rezultate bez renumeriranja. Cilj naredbe *renumberMesh* je poredati jednadžbe tako da se većina vandijagonalnih elemenata nalazi u samoj blizini dijagonale, u obliku dviju vrpce koje obuhvaćaju dijagonalu, čime se ubrzava rješavanje sustava jednadžbi. U slučaju AMG algoritma gdje je poredak jednadžbi bitan za uspješnost izvedbe algoritma ovisno o primijenjenom zaglađivaču, rješenja se mogu znatno razlikovati pri korištenju ILU algoritma kao zaglađivača kojemu zgusnutost elemenata oko dijagonale matrice mnogo više odgovara. U tablici 6.5 vidljivo je da je prije renumeriranja kontrolnih volumena simulacija za slučaj outlet primjenom ILU algoritma prekinuta prije dostizanja 1000. iteracije, dok je renumeracijom kontrolnih volumena (zgušnjavanjem elemenata na pozicije oko dijagonale) dostignuto rješenje prije 1000. iteracije, tablica 6.4. Iz navedenih rezultata daje se zaključiti kako je za ILU algoritam veoma bitan položaj elemenata u mreži te da mu specifično najviše odgovara zgusnutost elemenata oko dijagonale.

Međutim, kombinacija *strongest first* poretka jednadžbi s jednim i drugim zaglađivačem pokazala je drugačije rezultate na mreži s izduženim kontrolnim volumenima u smjeru osi x (mreža s 25520 kontrolnih volumena). U ovom slučaju kontrolni volumeni nisu uniformni, odnosno jednakih duljina svih stranica, kao u dosadašnjim korištenim mrežama već imaju izdužene stranice u smjeru osi x čije se dimenzije povećavaju od ulazne do izlazne plohe (objašnjeno u poglavlju 5.). Širenje informacija u ovom slučaju, u usporedbi s prethodnim mrežama, poboljšano je u smjeru okomitom na izdužene stranice jer sada kontrolni volumeni imaju veću površinu plohe preko koje komuniciraju, dok je otežano u smjeru okomitom na stranice koje su kraće. Gauss-Seidel algoritam traži što veću uniformnost kontrolnih volumena kako bi se informacije širile podjednako u svim smjerovima. Ako su kontrolni volumeni veoma izduženi u jednom smjeru, tada će u tom smjeru veza biti jaka i koeficijent u matrici velik, dok u smjeru u kojem nisu kontrolni volumeni nisu izduženi, veza će biti slaba i koeficijent u matrici mali. Prilikom dobivanja rješenja Gauss-Seidelom takve razlike u veličini koeficijenata uzrokovat će sporije dobivanje rješenja, odnosno sporiju konvergenciju. Ovdje se ILU algoritam pokazao boljom opcijom s obzirom na to da na njegov rad ne utječe neuniformnost kontrolnih volumena u mreži. ILU algoritam radi nepotpunu faktorizaciju kojom ostavlja popunjena samo ona mjesta koja su bila popunjena u finoj matrici, dok je sada poredak jednadžbi drugačiji, što znači da su drugi koeficijenti na mjestu onih iz fine matrice.

Ako je matrica zadržala uzorak vrpce oko dijagonale, točnije ako se stvaranjem grubih razina elementi nisu previše raspršili van dijagonale, tada ILU algoritam svojom faktORIZACIJOM ne ispušta velike elemente bitne za širenje informacija i konvergencija rezultata je bolja. Proces *coarsening*-a eliminira jednadžbe kao fine ako imaju svoju jaku vezu na gruboj razini. Kod izduženih kontrolnih volumena neke veze između susjeda postaju slabe naspram drugima i ostaju zadržane na gruboj razini čime ILU algoritam ima veće šanse da ne eliminira velik volumen prilikom faktorizacije i dolazi brže do rješenja.

Lokacije zadavanja vrijednosti tlaka

U simulacijama je korišteno zadavanje vrijednosti tlaka na različitim položajima unutar modela. Pokazano je da, što se konvergencije linearnog rješavača tiče, nema nikakve pravilnosti u brzini konvergencije ovisno o položaju zadavanja vrijednosti tlaka. Naime, multigrid algoritam tijekom stvaranja grubih razina eliminira jednadžbe (a svaka jednadžba predstavlja jedan kontrolni volumen) gdje se može dogoditi da se eliminira upravo ona jednadžba koja sadrži zadanu vrijednost tlaka, neovisno o tome gdje se taj kontrolni volumen nalazi, pa će konvergencija biti sporija iako je položaj kontrolnog volumena u kojem je zadan tlak potencijalno povoljniji (s obzirom na smjer strujanja) od nekog drugog. Jedina pokazana pravilnost koja vrijedi za sve testirane mreže jest smanjivanje razlika u brzini konvergencije reziduala s većim brojem jednadžbi na gruboj razini neovisno o varijanti zadavanja tlaka. Što je više jednadžbi na gruboj razini to fina matrica primi više korekcija i propagacija informacija je bolja zbog čega lokacija zadavanja tlaka prestaje imati važnu ulogu u brzini konvergencije rješenja. Ovaj trend smanjenja razlika u brzini konvergencije s većim brojem jednadžbi na gruboj razini manje je uočljiv kod najgušće mreže, 76800 kontrolnih volumena, jer promjena postavke broja jednadžbi na gruboj razini (5, 20 i 100) nije dovoljno značajna s obzirom na velik ukupni broj kontrolnih volumena. Može se izvući zaključak da kod gušćih mreža nije potrebno zadavati mali broj jednadžbi na gruboj razini, te da će konvergencija reziduala linearnog rješavača biti brža i manje ovisna o lokaciji zadavanja vrijednosti tlaka ako je broj jednadžbi na gruboj razini veći. Nadalje, zadavanje tlaka na čitavoj izlaznoj plohi, slučaj outlet, kontinuirano pokazuje najveću kumulativnu grešku kontinuiteta u odnosu na sve ostale slučajeve gdje je tlak zadan u jednom kontrolnom volumenu. Razlog leži u uniformnosti ovog rubnog uvjeta jer se na izlazu prirodno formira izobraženo strujanje čiji profil brzine ima oblik parabole - najveća brzina u središtu cijevi, nula na

stijenkama - stoga bi tlak trebao biti najveći na stijenkama (tamo gdje je brzina 0), a najmanji u središtu. Nametanjem fiksne vrijednosti tlaka na čitavoj izlaznoj plohi dobiva se nefizikalno rješenje zbog čega je i kumulativna greška kontinuiteta veća u odnosu na druge slučajeve gdje se strujanje na izlaznoj plohi formira slobodno.

Broj iteracija linearnog rješavača

Sljedeća primijećena stavka jest broj iteracija linearnog rješavača koji bi prema teoriji uz korištenje Gauss-Seidel algoritma kao zaglađivača i iste postavke AMG rješavača, ali na različitim mrežama, u istoj nelinearnoj iteraciji trebao biti jednak za svaku mrežu. Za sve tri testirane mreže konvergencije reziduala linearnog rješavača su prikazane u istoj nelinearnoj iteraciji, 1000.. Uspoređujući slike 6.3, 6.6 i 6.9 za slučaj zadavanja tlaka na izlaznoj plohi (outlet) može se primijetiti kako to nije slučaj, već za iste postavke za mrežu od 48000 volumena broj iteracija linearnog rješavača iznosi 17, za mrežu od 19200 kontrolnih volumena iznosi 22, a za mrežu od 76800 iznosi 43. Promotri li se drugi slučajevi zadavanja tlaka, također nema podudarnosti. S obzirom na to da različiti slučajevi lokacije zadavanja vrijednosti tlaka nisu pokazali pravilnost u brzini konvergencije, uzme li se cijeli raspon broja linearnih iteracija na promatranim dijagramima za usporedbu, i dalje se ne uočavaju podudaranja za sve tri mreže. Može se zaključiti kako se teorija o poklapanju rezultata u istim nelinearnim iteracijama za različite mreže s istim postavkama u praksi nije pokazala točnom.

7.2.2. *fine order* poredak jednadžbi

Gauss-Seidel algoritam naprotiv može raditi i s *fine order* poretkom jednadžbi na gruboj razini, no pokazano je da je konvergencija reziduala linearnog rješavača brža ako se na takvom poretku koristi ILU algoritam kao zaglađivač, slika 6.15. Razlog leži u načinu na koji Gauss-Seidel redom propagira kroz matricu i ako se jednadžbe s najviše utjecaja, u *strongest first* poretku, nalaze prve u matrici, tada će se njihovim rješavanjem utjecaj širiti jer jednadžbe koje dolaze na red kasnije uzimaju informacije od jednadžbi riješenih prije i što se više ažuriranih informacija (koje dolaze od jakih veza) koristi za rješavanje jednadžbe, to je vrijeme potrebno za doći do rješenja kraće, odnosno konvergencija brža. *Fine order* poredak jednadžbe s najviše utjecaja ostavlja na mjestima u kojima su se nalazile u finoj matrici zbog čega ih Gauss-Seidel kasnije

rješava i učinak njihovog utjecaja je znatno smanjen, a time i konvergencija sporija što je vidljivo iz rezultata.

Osvrne li se na rezultate s poretkom jednadžbi na gruboj razini *fine order* i korištenjem ILU algoritma kao zaglađivača, tablica 6.7, mogu se uočiti divergencija rješenja za sve tri postavke minimalnog broja jednadžbi na gruboj razini. Divergencije rješenja se mogu objasniti načinom rada ILU algoritma koji s kreiranjem svake sljedeće grube razine ponovo radi nepotpunu **LU** faktorizaciju prilikom koje izostavlja elemente iz matrice kako bi zadržao isti uzorak popunjenosti matrice kao na prethodnoj finoj razini. Na svakoj grubljoj razini matrica je sve gušće popunjena, a ILU algoritam nepotpunom faktorizacijom izostavlja elemente sve dok dolaskom na najgrublju razinu algoritam potencijalno ispusti velik broj jako utjecajnih elemenata zbog čega se informacije ne mogu širiti, točnije interpolirati na fine razine. Navedene tvrdnje mogu se proučiti u tablici 7.2 promatrajući broj elemenata na svakoj razini za slučajeve zadavanja tlaka na čitavoj izlaznoj plohi outlet i minimalni broj jednadžbi na gruboj razini 5, 20 i 100.

minCoarseEquations 5				
Razina	Ukupno mjesta u matrici	Vandijagonalni elementi	Praznih mjesta	Udio praznih mjesta [%]
1	5898240000	305760	5897857440	99.99
2	1474560000	286692	1474234908	99.98
3	175271121	136366	175121516	99.91
4	14424804	52258	14368748	99.61
5	938961	14114	923878	98.39
6	61009	3288	57474	94.21
7	4096	632	3400	83.01
8	400	126	254	63.50
minCoarseEquations 20				
Razina	Ukupno mjesta u matrici	Vandijagonalni elementi	Praznih mjesta	Udio praznih mjesta [%]
1	5898240000	305760	5897857440	99.99
2	1474560000	288590	1474233010	99.98
3	171950769	136524	171801132	99.91
4	14032516	50350	13978420	99.61
5	923521	13482	909078	98.44
6	59536	3118	56174	94.35
7	4096	644	3388	82.71
minCoarseEquations 100				
Razina	Ukupno mjesta u matrici	Vandijagonalni elementi	Praznih mjesta	Udio praznih mjesta [%]
1	5898240000	305760	5897857440	99.99
2	1474560000	289538	1474232062	99.98
3	172186884	137000	172036762	99.91
4	14160169	50728	14105678	99.62
5	917764	13414	903392	98.43
6	64009	3366	60390	94.35

Tablica 7.2: Tablica usporedbe broja elemenata u matrici

U tablici 7.2 može se primijetiti da je za sve postavke minimalnog broja jednadžbi na gruboj razini udio praznih mjesta u matrici otprilike podjednak do 6. razine kada u oba slučaja iznosi oko 94%. Navedeni udio praznih mjesta znači da postoje manje šanse da se veliki elementi nalaze raspršeni u matrici, na mjestima udaljenijim od dijagonale. Promotri li se slučaj sa zadanim minimalnim brojem jednadžbi na gruboj razini 5, primjetno je da je broj stvorenih razina veći od postavke minimalnog broja jednadžbi 20 i može se uočiti udio praznih mjesta koji na 7. i 8. razini znatno opada u odnosu na prethodne razine. Opadanje udjela praznih mjesta ukazuje na povećanje gustoće matrice dolaskom na grube razine zbog čega se sve veći broj elemenata nalazi na mjestima u matrici udaljenijim od dijagonale. ILU algoritam nepotpunom faktorizacijom zadržava samo elemente koji se nalaze na istim mjestima kao u finoj matrici zbog čega se s većom

popunjenosti matrice sve veći broj elemenata izostavlja. Na najgrubljim razinama zbog ispuštanja velikog broja utjecajnih elemenata dolazi do velikih iznosa reziduala što može dovesti do divergencije rješenja kako je prikazano u tablici 6.7. U tablici 6.7 primjećuje se da ni uvjet od minimalnog broja jednadžbi na gruboj razini 100 s obzirom na veličinu problema nije bio dovoljno velik da ILU algoritam ne ispusti previše bitnih elemenata. Zaključno, pri korištenju ILU algoritma pokazano je da velik broj grubih razina, odnosno premali broj jednadžbi na gruboj razini, može uzrokovati divergenciju rješenja zbog čega se treba obratiti posebnu pozornost na postavke pri korištenju ILU algoritma.

Rezultati na tetraedarskim mrežama

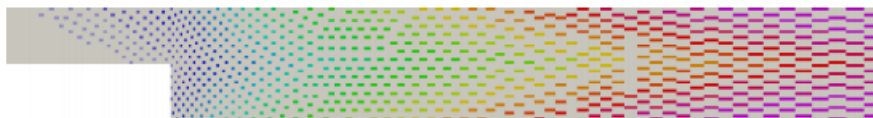
ILU algoritam testiran je na tetraedarskim mrežama nakon što je u prethodnim simulacijama uočeno kako oblik kontrolnih volumena nije imao velik značaj na rad ILU algoritma kao zaglađivača. Prvo su testirane i uspoređene dvije tetraedarske mreže, jedna s rasporedom gustoće kontrolnih volumena kao na prethodnim mrežama i druga s povećanim kontrolnim volumenima na izlazu iz cijevi većeg promjera. Utvrđeno je kako ne samo da ILU algoritam funkcionira na tetraedarskoj mreži, već da nagla promjena u veličini kontrolnih volumena usred proračunske mreže nije imala gotovo nikakav značajni utjecaj na rad algoritma. Ovime se dokazalo da oblik kontrolnih volumena (izduženost u jednom ili drugom pravcu, veličina, itd.) za ILU algoritam zaista nemaju veliki značaj što znači da ILU algoritam dobro funkcionira i na mrežama s veoma grubim prijelazima u gustoći i veličini kontrolnih volumena, dok je Gauss-Seidel pokazao usporavanje konvergencije već pri samom izduživanju kontrolnih volumena na izlazu cijevi većeg promjera. Broj kreiranih grubih razina je smanjen kod mreže s povećanim kontrolnim volumenima na izlazu zbog manjeg broja kontrolnih volumena mreže (manjeg broja jednadžbi u matrici).

Uspoređeni su i rezultati dobiveni na heksaedarskoj i tetraedarskoj mreži kako bi se donijeli bolji zaključci o radu ILU algoritma. Pokazalo se da je broj kreiranih grubih razina manji kod tetraedarske mreže zbog manjeg broja kontrolnih volumena mreže u odnosu na heksaedarsku što je pokazano u svim dosadašnjim simulacijama. Zanimljivost u ovim rezultatima je bila značajna razlika u broju iteracija linearnog rješavača gdje je kod tetraedarske mreže taj broj za pojedine slučajeve dvostruko veći u usporedbi s heksaedarskom mrežom. Govoreći samo o broju kontrolnih volumena u mrežama, u dosadašnjim rezultatima pokazalo se da se za rjeđe i gušće mreže broj iteracija linearnog

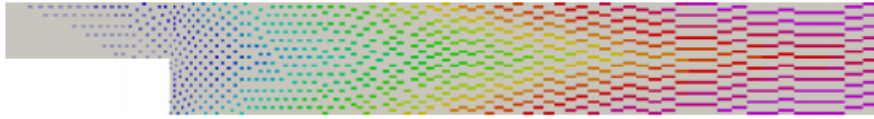
rješavača razlikuje, ali uglavnom se radi o samo nekoliko iteracija više. Kod tetraedarske mreže taj broj je puno veći što se može pripisati manjem broju ploha preko kojih kontrolni volumeni komuniciraju zbog čega je širenje informacija sporije u odnosu na heksaedarsku mrežu. Ovaj zaključak generalno nema veze sa samim ILU algoritmom, već s prirodom testiranih mreža.

7.2.3. Snaga veza između jednadžbi

Dobiveni rezultati s promjenom snage veze pokazali su kako povećanje snage veze s $\varepsilon = 0.2$ na $\varepsilon = 0.3$ nije imalo nikakav utjecaj na dobivena rješenja, neovisno o algoritmu korištenom u ulozi zaglađivača. Ovakvi rezultati mogu se objasniti činjenicom da su simulacije provedene na 2D mreži. Naime, snaga veze utječe na to koliko će jednadžbi nakon procesa *coarsening*-a ostati na gruboj razini. Koeficijenti u matrici proporcionalni su površinama dodira susjednih kontrolnih volumena. Tako je opisana snaga veze između volumena. Povećavanjem vrijednosti snage veze povećava se uvjet odabira jakih veza jer sada ostali koeficijenti jednadžbe moraju biti veći od umnoška novog ε i najvećeg koeficijenta kako bi bili sortirani u jake veze. S obzirom na to da isti koeficijenti koji su s uvjetom snage veze $\varepsilon = 0.2$ bili priznati kao jake veze, s novim uvjetom više nisu, to znači da će više jednadžbi morati ostati na gruboj razini jer nemaju svoju jaku vezu iz koje se mogu interpolirati vrijednosti s grube na finu razinu putem matrice prolongacije. Jednadžbe koje opisuju susjedne kontrolne volumene zato ostaju na gruboj razini kako bi se omogućilo širenje informacija između tih volumena, ali istovremeno se širenje informacija ograničava prema ostalim kontrolnim volumenima jer druge jednadžbe u zamjenu postaju fine i za njih se informacija interpolira iz manjeg broja kontrolnih volumena nego što je to bio slučaj s manjom vrijednosti snage veze. Navedene promjene uzrokuju usporavanje konvergencije ili čak divergenciju reziduala zbog nemogućnosti širenja informacija. Na slikama 7.1 i 7.2 za usporedbu su prikazani izgledi grube razine s vrijednostima snage veze 0.1 i 0.3 preuzetim iz [2]:



Slika 7.1: Preostale jednadžbe na gruboj razini sa snagom veze 0.1



Slika 7.2: Preostale jednadžbe na gruboj razini sa snagom veze 0.3

Na slikama su bojama označene jednadžbe (koje predstavljaju kontrolne volumene na finoj razini) zadržane na gruboj razini dok su sivom bojom prikazane jednadžbe eliminirane kao fine. Može se uočiti da je raspored odabranih jednadžbi na gruboj razini raspršeniji na slici 7.1, kao šahovska ploča, što omogućava da jednadžbe na finoj razini interpolacijom dobiju utjecaj od više susjeda (veza) čime se ubrzava širenje informacija (i time konvergencija rezultata). Na slici 7.2, posebice prema izlaznoj plohi, uočavaju se područja gdje su jednadžbe susjednih kontrolnih volumena zadržane na gruboj razini, prikazani duljim obojenim prugama koje su zapravo niz susjednih kontrolnih volumena. Naspram njima uočavaju se i veća područja sive boje koja predstavljaju kontrolne volumene čije su jednadžbe odabrane kao fine. Zbog takvog rasporeda će jednadžbe koje pripadaju sivom području (fine jednadžbe) dobiti manje informacija iz susjedstva no što dobivaju u slučaju manje vrijednosti snage veze, slika 7.1, i rješavanje sustava jednadžbi se usporava ili čak prekida zbog prevelikih vrijednosti reziduala koji divergira umjesto konvergira.

Objašnjeni slučaj odnosi se na 3D mrežu gdje se utjecaj ostavljanja jednadžbi susjednih kontrolnih volumena vidi zbog izduženosti u treću dimenziju prostora. Prikaz na slici 2, dulje obojene pruge koje predstavljaju susjedne kontrolne volumene čije su jednadžbe zadržane na gruboj razini, u 2D slučaju koji je simuliran u ovom radu nema vidljiv utjecaj jer je protok informacija kroz plohe znatno smanjen zbog minimalne izduženosti volumena u smjeru treće komponente prostora. Zbog toga naveden promjene koje bi se u 3D slučaju uočile na rezultatima, u prikazanom 2D slučaju nisu vidljive.

7.2.4. Promjena sume elemenata u retku matrice prolongacije

Sljedeća postavka koja je pokazala utjecaj na konvergenciju rješenja jest promjena sume elemenata u retku matrice prolongacije. Uočena je smanjena brzina konvergencije reziduala linearnog rješavača u usporedbi s istim postavkama AMG rješavača kada suma redaka matrice prolongacije nije postavljena u 1, već se formira s obzirom na

karakteristike jednadžbe u finoj matrici. Nadalje, neki slučajevi u tablici 6.11 pokazuju i divergenciju rješenja prije dostizanja 1000. iteracije za koju su rezultati prikazani. Razlog ovakvim rezultatima je to što se nametanjem sumacije redaka matrice prolongacije u 1, umjetno stvaraju interpolacijski faktori matrice prolongacije koji ne odgovaraju karakteristikama jednadžbi u finoj matrici. Potom se, nakon formiranja matrice prolongacije, gruba matrica linearnog sustava \mathbf{A}^C formira kao Galerkinova matrica (potpoglavlje 3.4.2.): $\mathbf{A}^C = \mathbf{R}\mathbf{A}^F\mathbf{P}$, gdje se fina matrica množi novoformiranom matricom prolongacije \mathbf{P} , a posljedično i matricom restrikcije \mathbf{R} . Upravo tim množenjem s matricama prolongacije i restrikcije jednadžbe koje su bile dijagonalno dominantne na finoj razini na gruboj više nisu. Jednadžbe koje su dijagonalno dominantne doprinose brzom konvergenciji rješenja, odnosno smanjuju broj iteracija potrebnih da se dođe do željene točnosti rješenja, te se uklanjanjem njihove dominantnosti otežava širenje informacija kroz matricu i usporava rješavanje sustava jednadžbi. Zbog toga se u prikazanim rezultatima neke simulacije i prekidaju, odnosno ne mogu doseći rješenje jer prilikom rješavanja rezidual divergira u besmisleno velike vrijednosti.

Opisano smanjivanje broja dijagonalno dominantnih jednadžbi prikazano je u tablici 6.12 u donjem dijelu tablice. U slučaju gdje sumacija redaka matrice prolongacije u 1 nije nametnuta, gornji dio tablice, vidljivo je da se sa smanjivanjem broja jednadžbi na grubim razinama broj dijagonalno dominantnih jednadžbi smanjuje zbog smanjivanja ukupnog broja jednadžbi i u mnogo manjoj mjeri no što se smanjuje kod nametnute sumacije redaka u 1. Navedena razlika nastaje zbog promjene karakteristika određenog broja dijagonalno dominantnih jednadžbi, odnosno one zbog množenja s izmijenjenom matricom prolongacije gube dijagonalnu dominantnost.

7.2.5. Broj primjene zaglađivača prije i nakon AMG rješavača

Ovo je postavka koja je pokazala značajnu razliku u odnosu na teoriju multigrid algoritma. Teorija nalaže da broj primjene algoritma zaglađivača prije i nakon ciklusa AMG rješavača mora biti jednak, odnosno da čitav ciklus mora biti simetričan - na primjer $\mathbf{S}^2\mathbf{M}\mathbf{S}^2$ - kako bi se sačuvale svojstvene vrijednosti matrice linearnog sustava i osiguralo funkcioniranje multigrid algoritma. Testiranjem drugačijih postavki, odnosno različitog broja *pre-smoothing* i *post-smoothing* procesa, pokazano je da ne samo da broj primjene zaglađivača prije i nakon AMG rješavača može biti različit, već da postoje i

određene kombinacije koje osiguravaju bržu konvergenciju od simetričnog ciklusa. Iz rezultata prikazanih na dijagramu 6.20 može se zaključiti kako je za bržu konvergenciju reziduala potrebno postaviti veći broj ponavljanja *post-smoothing* procesa u odnosu na *pre-smoothing*. Za slučaj kada je zadana vrijednost tlaka na čitavoj izlaznoj plohi (slučaj outlet) konvergencija je bila najbrža za slučaj s 1 *pre-smoothing* i 3 *post-smoothing* procesa, dok se za slučaj zadavanja tlaka u samo jednom kontrolnom volumenu (slučaj KV outlet) pokazala najbolja kombinacija s 2 *pre-smoothing* i 2 *post-smoothing* procesa. Brža konvergencija s većim brojem *post-smoothing* procesa može se objasniti zaglađivanjem rješenja nakon primjene korekcije koje onda ima značajniju ulogu u sljedećoj nelinearnoj iteraciji gdje iteracija kreće s uklonjenim većim brojem komponenti grešaka iz prethodnog koraka. Na taj način se kompenzira velika kumulativna greška kontinuiteta prilikom zadavanja vrijednosti tlaka na čitavoj izlaznoj plohi outlet. *Pre-smoothing* služi kao priprema fine matrice za proces *coarsening*-a i rješavanja sustava AMG rješavačem. Zbog prirode ILU algoritma koji ispušta elemente tijekom faktorizacija i mogućnosti da ispusti upravo onaj element u kojem je zadana vrijednost tlaka čime se unosi greška, slučaj KV outlet, veći broj *pre-smoothing* procesa pomaže u kompenzaciji greške i rezultira bržom konvergencijom.

7.3. Sažetak rezultata

U ovom potpoglavlju bit će tablično navedeni zaključci doneseni o pojedinim algoritmima zaglađivačima i općenito o multigrid algoritmu. Zaključci su sažeti kako bi se budućim korisnicima mogao omogućiti brzi pregled utjecaja i odabir primjerenih postavki.

Tablica 7.3 prikazuje zaključke donesene za rad algoritma zaglađivača Gauss-Seidel i ILU ovisno o zadanim postavkama.

Gauss-Seidel	Postavka/zaključak	Utjecaj na rezultate
Poredak jednadžbi na gruboj razini	strongest first	brza konvergencija
	fine order	spora konvergencija
Broj jednadžbi na gruboj razini	funkcionira neovisno o broju jednadžbi	-
Oblik kontrolnih volumena	uniformni kontrolni volumeni	brza konvergencija
	neuniformni kontrolni volumeni	spora konvergencija
Identitet kontrolnih volumena	renumberMesh	nema značajan utjecaj na konvergenciju

ILU	Postavka/zaključak	Utjecaj na rezultate
Poredak jednadžbi na gruboj razini	strongest first	divergencija rješenja ili brza konvergencija ovisno o uvjetu zadavanja tlaka
	fine order	konvergencija ovisna o broju jednadžbi na gruboj razini
Broj jednadžbi na gruboj razini	pre mali broj jednadžbi	divergencija rješenja
Oblik kontrolnih volumena	nema ograničenja	-
Identitet kontrolnih volumena	renumberMesh	ubrzanje konvergencije

Tablica 7.3: Tablica sažetka postavki algoritama zaglađivača

Tablica 7.4 prikazuje zaključke donesene o radu multigrid algoritma ovisno o zadanim postavkama.

Multigrid algoritam	Postavka/zaključak	Utjecaj na rezultate
Lokacija zadavanja tlaka	outlet	Kumulativna greška kontinuiteta veća
	jedan KV u mreži - neovisno gdje se nalazi	Kumulativna greška kontinuiteta manja
Broj primjene zaglađivača	veći broj post-smoothing procesa (1-3, 2-3)	brža konvergencija
	veći broj pre-smoothing procesa (3-1)	sporija konvergencija
	simetrično (2-2)	srednje brza konvergencija
Snaga veze između jednadžbi	veća/manja snaga veze	nema utjecaja za 2D mreže
Promjena sume elemenata u retku matrice u prolomgacije	smanjuje broj dijagonalno dominantnih jednadžbi	divergencija rješenja

Tablica 7.4: Tablica sažetka postavki multigrid algoritma

7.4. Zaključak

U ovom poglavlju provedena je diskusija o dobivenim rezultatima. Iznesena su detaljna objašnjenja rezultata dobivenih za različite mreže s istim postavkama. Objasnjeni su rezultati dobiveni poretkom jednadžbi *fine order* i *strongest first* za oba algoritma zaglađivača, Gauss-Seidel i ILU, te rezultati dobiveni promjenom snage veze između jednadžbi i promjenom sume elemenata u retku matrice prolongacije. Doneseni zaključci sažeti su u tablice na kraju poglavlja kako bi se dao jednostavniji prikaz utjecaja postavki na konvergenciju rezultata.

U sljedećem poglavlju bit će dan kratak pregled rada i donesenih zaključaka.

8 | Zaključak

8.1. Zaključak

Ovaj rad bavi se performansama multigrad algoritma koji omogućava eliminaciju komponenti grešaka koje algoritmi fiksne točke ne mogu efikasno eliminirati. Na samom početku rada opisan je linearni sustav jednadžbi koji opisuje modelirani problem strujanja. Potom su opisani algoritmi stacionarne točke za rješavanje linearnog sustava, Gauss-Seidel i Gaussove eliminacije, iz čijih se nedostataka izvodi motivacija za korištenje multigrad algoritma. Opisan je i ILU algoritam kao još jedan od algoritama za rješavanje linearnog sustava koji se kasnije u radu, uz Gauss-Seidel algoritam, koristi kao zaglađivač u multigrad ciklusu. Slijedi detaljno objašnjenje rada multigrad algoritma i njegove postavke u rječniku *fvSolution* unutar *foam-extend-a* koje se primjenjuju na različite mreže objašnjene narednom poglavlju.

Rezultati provedenih simulacija prikazani su ovisno o dva različita poretka jednadžbi na gruboj razini multigrad algoritma - *strongest first* i *fine order* poredak - te o algoritmu koji se primjenjuje kao zaglađivač u multigrad ciklusu - Gauss-Seidel ili ILU. Mreže nad kojima su simulacije provedene različitih su gustoća, odnosno broja kontrolnih volumena, te oblika kontrolnih volumena (izduženi kontrolni volumeni u pojedinim područjima geometrije, kontrolni volumeni u obliku heksaedara ili tetraedara, veće dimenzije kontrolnih volumena, i slično). Testirane postavke multigrad algoritma su broj jednadžbi na gruboj razini, broj *pre-smoothing* i *post-smoothing* procesa te promjena algoritma zaglađivača. Ove postavke mijenjane su unutar rječnika *fvSolution* dok su postavke snage veze između jednadžbi i promjena sume elemenata u retku matrice prolongacije mijenjane u izvornom kodu multigrad algoritma. Također, mijenjane su lokacije

zadavanja fiksne vrijednosti tlaka, od tlaka zadanog na čitavoj izlaznoj plohi do tlaka zadanog u samo jednom kontrolnom volumenu na različitim lokacijama unutar mreže.

Promatrajući rezultate dobivene korištenjem Gauss-Seidel algoritma na poretku jednadžbi *strongest first*, uočeno je da algoritam radi bez divergencije rješenja za sve isprobane postavke broja jednadžbi na gruboj, za sve lokacije zadavanja tlaka. Koristi li se Gauss-Seidel algoritam na poretku jednadžbi *fine order*, uočena je sporija konvergencija rezultata iako i dalje bez divergencije rješenja. Primjenom ILU algoritma rezultati su nešto drugačiji, s pojavama divergencije rješenja kod oba poretka jednadžbi. Na *strongest first* poretku divergencije se mogu objasniti načinom na koji ILU algoritam rješava sustav jednadžbi gdje na takvom poretku jednadžbi ILU algoritam unosi grešku s rješavanjem svake nove grube razine matrice. Promatrajući divergencije rješenja na *fine order* poretku, objašnjenje se može pronaći u premalom broju jednadžbi na gruboj razini koje uz izostavljanje elemenata prilikom nepotpune faktorizacije dovodi do gubitka prevelikog broja informacija potrebnih za rješavanje sustava i rješenje divergira.

Nadalje, rješenja dobivena na mreži s izduženim kontrolnim volumenima pokazala su bolju konvergenciju korištenjem ILU algoritma u usporedbi s Gauss-Seidel algoritmom čime je pokazano da je za dobar rad Gauss-Seidel algoritma potrebna uniformnost kontrolnih volumena dok ILU algoritmu oblik kontrolnih volumena ne uvjetuje rad. Ova tvrdnja provjerena je i na tetraedarskim mrežama gdje je ILU algoritam, na *fine order* poretku jednadžbi, pokazao konvergenciju k rješenjima.

Potom je u izvornom kodu mijenjana postavka snage veze između jednadžbi s $\varepsilon = 0.2$ na $\varepsilon = 0.3$ što bi na 3D mrežama uzrokovalo značajno smanjenje konvergencije reziduala prema [2], no u slučaju 2D mreže u ovom radu, rezultati su bili u potpunosti jednaki rezultatima dobivenim za snagu veze $\varepsilon = 0.2$ za oba algoritma zaglađivača.

Posljednja promijenjena postavka jest sumiranje redova matrice prolongacije u 1. Prema dobivenim rezultatima može se uočiti da prisilnim sumiranjem redaka matrice prolongacije u 1 jednadžbe koje su prije bile dijagonalno dominantne, prestaju biti, čime se smanjuje brzina konvergencije, a u nekim slučajevima dolazi do divergencije rješenja.

Uzimajući u obzir sve gore navedeno, multigrid algoritam ima značajnu ulogu u smanjivanju greški i time ubrzavanju konvergencije rješenja za dominantno difuzijske sustave koji su veoma osjetljivi i dugotrajni za rješavanje, ali njegove postavke se trebaju detaljno proučiti i pravilno postaviti kako bi se izbjegle divergencije rješenja i pretjerane varijacije postavki zbog nepoznavanja prirode rada algoritma. Svaki od pri-

mijenjenih zaglađivača ima svoje prednosti i nedostatke koji se moraju uzeti u obzir prilikom proučavanja promatranog problema kako bi se odabrao najpovoljniji algoritam i ubrzala konvergencija rješenja. Ovaj rad namijenjen je korisnicima multigrid algoritma koji žele ući u dubinu razumijevanja njegovog rada i izvući maksimum njegovim korištenjem.

Literatura

- [1] I. Džijan Z. Virag. *Računalna dinamika fluida*. Fakultet strojarstva i brodogradnje, Zagreb, 2014.
- [2] Tessa Uroić. *Implicitly Coupled Finite Volume Algorithms*. PhD thesis, University of Zagreb, 2019.
- [3] C .A. J. Fletcher. *Computational Techniques for Fluid Dynamics 2*. Springer, 1988.
- [4] T.Uroić, H. Jasak, and H. Rusche. Implicitly Coupled Pressure-Velocity Solver. *OpenFOAM*, 2019.
- [5] S. Patankar and D. Spalding. *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows*. International journal of heat and mass transfer 15, (1972) 1787-1806.
- [6] H. Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows*. PhD thesis, Imperial College of Science, Technology and Medicine, 1996.
- [7] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2000.
- [8] U. Trottenberg, C. Oosterlee, and A. Schüller. Multigrid. *Elsevier and Academic Press*, 2001.
- [9] T.Uroić and H. Jasak. Parallelisation of selective algebraic multigrid for block-pressure-velocity system in OpenFOAM. *Computer Physics Communications*, 258, 2021.

- [10] T.Uroić and H. Jasak. Block-selective algebraic multigrid for implicitly coupled pressure-velocity system. *Computers & Fluids*, 167:100–110, 2018.
- [11] G. Wittum. *Linear iterations as smoothers in multigrid methods: theory with applications to incomplete decompositions, Impact of computing in science and engineering 1*. SIAM Journal on Scientific and Statistical Computing, 1986.