

Razvoj automatskog algoritma za pronalazak receptora kod preklapajućih mreža u Foam-Extend softveru

Katavić, Josip

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:199044>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2024-11-29**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



UNIVERSITY OF ZAGREB
FACULTY OF MECHANICAL ENGINEERING AND NAVAL
ARCHITECTURE

MASTER'S THESIS

Josip Katavić

ZAGREB, 2018

UNIVERSITY OF ZAGREB
FACULTY OF MECHANICAL ENGINEERING AND NAVAL
ARCHITECTURE

MASTER'S THESIS

DEVELOPMENT OF AUTOMATIC FRINGE ASSEMBLY ALGORITHM FOR
OVERSET MESHES IN FOAM-EXTEND SOFTWARE

Supervisor:

Prof. Hrvoje Jasak, PhD

Student:

Josip Katavić

ZAGREB, 2018

I would like to thank my supervisor, Professor Hrvoje Jasak, for giving me an opportunity to work on advanced CFD problems as a part of his group. Thank you for delivering interesting and unconventional lectures, which made CFD problems more clear and interesting to me. Thank you for your encouragement and honest advice.

I am pleased to give thanks to my unofficial supervisor, postdoctoral researcher Vuko Vukčević. Thank you for your time, guidance, constructive discussions, advice and help. It was a rewarding experience to work in a friendly atmosphere with an expert like you. Thank you for showing me by example what a young and successful engineer can achieve.

I am grateful to the academics Professor Ivo Senjanović and Dr. Marko Tomić for introducing me to sideways ship launching theory and for providing me with the articles, numerical and experimental data I used to validate the developed algorithm and the Naval Hydro Pack.

I am grateful to all my friends and colleagues, with whom I have spent nice moments during my studies and who were with me during the hard ones. Thank you for helping me create so many extraordinary moments I will never forget and will always cherish.

Finally, I am deeply grateful to my family for all the love and support they have provided me over the years. Thank you for your encouragement and for showing me the importance of education. Special thanks go to my aunts Marija Alajbegović and Nina Glavina, as well as to my uncle Krešimir Okmažić, who have been my support since my mother passed away.

Dedicated to my parents,
late Ksenija and Berislav

Dear mother,

I miss you often...

Thank you for everything!

Sit tibi terra levis.

Your son

Statement | Izjava

I hereby declare that I have made this thesis on my own using the knowledge acquired during my studies and the cited references.

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

Zagreb, June 2018.

Josip Katavić



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
procesno-energetski, konstrukcijski, brodstrojarski i inženjersko modeliranje i računalne simulacije

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **Josip Katavić**

Mat. br.: 0035190975

Naslov rada na hrvatskom jeziku: **Razvoj automatskog algoritma za pronalazak receptora kod preklapajućih mreža u Foam-Extend softveru**

Naslov rada na engleskom jeziku: **Development of Automatic Fringe Assembly Algorithm for Overset Meshes in Foam-Extend Software**

Opis zadatka:

Computational Fluid Dynamics (CFD) is used to solve a vast range of engineering problems for increasingly complex geometries. With increase in computer power and availability, the trends in CFD shift from steady state to transient computations. Transient computations often include moving objects such as: ships in waves, compressors, turbines and other. One of the often used methods for handling moving objects in CFD is the dynamic overset mesh approach, where two or more computational meshes are overlaid on top of each other. The interpolation of the field data between the meshes happens through the fringe layer, which must be determined in a robust and reliable way. Although the fringe layer can be manually selected, this represents an often unnecessary overhead from engineer's perspective. The goal of this Thesis is to develop an automatic algorithm for fringe assembly, which includes finding the most suitable set of acceptors in a reliable way. The candidate shall perform the following tasks:

- Perform a literature review for automatic fringe assembly algorithms as used in different commercial and scientific CFD codes.
- Review and familiarize himself with available fringe assembly strategies in Foam-Extend.
- Based on the available literature and existing methodology, propose an improved fringe assembly algorithm.
- Implement the proposed version of the fringe assembly algorithm.
- Test the newly developed overlap fringe assembly algorithm on three test cases:
 - o Two-levels mesh of cylinder in channel,
 - o Three-levels mesh of NACA 4412 profile,
 - o Six-levels mesh of ONRT ship with two rudders and propellers.
- Perform CFD simulation of side-ways ship launching with prescribed motion and compare the results with available experimental and numerical results

The Thesis shall list the bibliography and any assistance received during this study.

Zadatak zadan:

Datum predaje rada:

Predviđeni datum obrane:

3. svibnja 2018.

5. srpnja 2018.

11., 12. i 13. srpnja 2018.

Zadatak zadao:

Predsjednica Povjerenstva:

Prof. dr. sc. Hrvoje Jasak

Prof. dr. sc. Tanja Jurčević Lulić

Contents

Contents	vi
List of Figures	ix
List of Tables	xiii
List of Symbols	xv
Sažetak	xviii
Summary	xix
Prošireni sažetak	xx
1. Introduction	1
1.1. Overset Mesh	1
1.2. An Overview of Overset Mesh Codes	3
1.3. Launching Theory of Ships from Horizontal Berth by Tipping Table Arrangement	3
1.3.1. Launching Phases	6
1.4. Thesis Outline	12
2. Overset Mesh Technique	14
2.1. Introduction	14
2.2. Overset Mesh Cell Types	15

2.3.	Donor/Acceptor Pairs	18
2.4.	Overset Regions	19
2.5.	Overlap Fringe Assembly Strategies	20
2.5.1.	The <code>calcAddressing</code> Procedure	21
2.6.	Donor Suitability	22
2.7.	The <code>overlap</code> Algorithm	24
2.7.1.	Disadvantages of the <code>overlap</code> Algorithm	25
2.8.	The <code>adaptiveOverlap</code> algorithm	26
2.9.	Overset Interpolation Schemes	32
2.10.	Closure	33
3.	Overset Assembly Test Cases	34
3.1.	Two-Levels Mesh of Cylinder In a Channel	34
3.1.1.	Static Overset Assembly Test Case Setup 1 (standard grading) . .	36
3.1.2.	Static Overset Assembly Test Case Setup 2 (standard grading) . .	38
3.1.3.	Static Overset Assembly Test Case Setup 3 (inverse grading) . . .	41
3.1.4.	Dynamic Overset Assembly Test	42
3.2.	Three-Levels Mesh of NACA 4412 Airfoil	43
3.2.1.	Test Case Setup 1	45
3.2.2.	Test Case Setup 2	52
3.3.	Six-Levels Mesh of ONR Tumblehome Ship	56
3.4.	Closure	59
4.	Numerical Simulation of Side Launching of a 260 000 DWT Tanker by Tipping Table Principle	61
4.1.	Introduction	61
4.2.	Numerical Case Setup	62
4.2.1.	Computational Domain	62
4.2.2.	Finite Volume Meshes	65
4.2.3.	Boundary Conditions	67
4.2.4.	Overset Mesh Setup	68
4.2.5.	Prescribed Motions	69
4.2.6.	Simulation Properties	73
4.2.7.	Results and Discussion	74

4.2.8. Closure	77
5. Conclusion	78
Bibliography	80

List of Figures

1	Tipovi ćelija preklopljenih mreža.	xxii
2	Četvero-razinski sustav preklopljenih mreža.	xxiv
3	Dijagram iterativnog procesa.	xxix
4	Dijagram toka algoritma (prvi dio).	xxxi
5	Dijagram toka algoritma (drugi dio).	xxxii
6	Pozadinska i overset mreža s detaljnim pogledom (standardna gradacija).	xxxv
7	Polja tipova overset ćelija.	xxxvii
8	Konačni sustav preklopljenih mreža.	xxxvii
9	Pozadinska i overset mreža s detaljnim pogledom (obratna gradacija).	xxxviii
10	Polja tipova overset ćelija.	xxxix
11	Konačni sustav preklopljenih mreža.	xxxix
12	Prikaz sustava preklopljenih mreža u početnom trenutku	xl
13	Uzdizanje slobodne površine na lokaciji mjerenja	xli
1.1	Launching system [1].	4
1.2	Launching system in phase 0 [2].	6
1.3	Launching system in phase 1 [2].	7
1.4	Transition from phase 1 to phase 2 [2].	8
1.5	Launching system in phase 3 [2].	9
1.6	Launching system in phase 4 [2].	10
1.7	Launching system in phase 5 [2].	11
1.8	Launching system in phase 6 [2].	12

2.1	Overset mesh cell types.	15
2.2	Interpolation stencil.	16
2.3	Four-levels overset mesh system.	17
2.4	Fringe algorithms class diagram.	21
2.5	Iterative procedure diagram.	28
2.6	Flow chart of the <code>adaptiveOverlap</code> (part 1).	30
2.7	Flow chart of the <code>adaptiveOverlap</code> (part 2).	31
3.1	Geometry of the computational domain for the 2-D cylinder in a channel test case.	35
3.2	Combined meshes with detailed view (standard grading).	35
3.3	Combined meshes with detailed view (inverse grading).	35
3.4	Overset types.	37
3.5	Final overset assembly.	37
3.6	Overset types.	40
3.7	Final overset assembly detailed view.	40
3.8	Iteration 1.	41
3.9	Final assembly.	42
3.10	Overset mesh assembly at $t = 0s$	42
3.11	Overset mesh assembly at $t = 0.15s$	42
3.12	Overset mesh assembly at $t = 0.3s$	43
3.13	Overset mesh assembly at $t = 0.45s$	43
3.14	NACA 4412 overset meshes.	44
3.15	Zoomed view of NACA 4412 overset meshes.	44
3.16	Iteration 1 for <i>backgroundMesh</i> region.	46
3.17	Iteration 2 for <i>backgroundMesh</i> region.	46
3.18	Overset types field for <i>backgroundMesh</i> region.	47
3.19	Iteration 1 for <i>middleMesh</i> region.	48
3.20	Iteration 2 for <i>middleMesh</i> region.	48
3.21	Iterations 2 and 3 for <i>middleMesh</i> region (full view).	48
3.22	Overset types field for <i>middleMesh</i> region.	49
3.23	Iteration 1 for <i>frontMesh</i> region.	50
3.24	Iteration 2 for <i>frontMesh</i> region.	50

3.25	Iteration 3 for <i>frontMesh</i> region.	50
3.26	Iteration 4 for <i>frontMesh</i> region.	51
3.27	Overset types field for <i>frontMesh</i> region.	51
3.28	Final overset mesh assembly.	52
3.29	Overset types field for <i>backgroundMesh</i> region.	54
3.30	Overset types field for <i>middleMesh</i> region.	54
3.31	Overset types field for <i>frontMesh</i> region.	54
3.32	Detailed view of final overset mesh assembly.	55
3.33	Different views of ONR Tumblehome ship.	56
3.34	Overset types field for <i>propellerStarboard/propellerPort</i> regions.	58
3.35	Overset types field for <i>rudderStarboard/rudderPort</i> regions.	58
3.36	Overset types field for <i>background</i> region.	59
3.37	Overset types field for <i>hull</i> region.	59
4.1	Water basin with ship model on tipping table [3].	63
4.2	Meshes arrangement and free surface position at initial moment.	64
4.3	Dimensions of <i>backgroundMesh</i> region.	64
4.4	Dimensions of <i>boatMesh</i> region.	65
4.5	Full view of the <i>backgroundMesh</i> region.	65
4.6	Detailed view of the <i>backgroundMesh</i> region around initial free surface position.	66
4.7	Full view of the <i>boatMesh</i> region.	66
4.8	Combined meshes at initial moment of side launching.	67
4.9	Computational domain with patch IDs marked.	67
4.10	Horizontal displacement (experimental data).	70
4.11	Vertical displacement (experimental data).	70
4.12	Roll angle (experimental data).	71
4.13	Prescribed horizontal and vertical displacement.	72
4.14	Prescribed roll motion.	73
4.15	Phase fraction field at $t = 0$ s.	74
4.16	Phase fraction field at $t = 15$ s.	74
4.17	Phase fraction field at $t = 19$ s.	74
4.18	Phase fraction field at $t = 21.3$ s.	75

4.19 Phase fraction field at $t = 28.9$ s.	75
4.20 Phase fraction field at $t = 35.9$ s.	75
4.21 Surface elevation.	76

List of Tables

1	Parametri algoritma.	xxvii
2	Vrijednosti volumnoga polja tipova ćelija.	xxx
3	Relevantni podaci iteracija.	xxxvi
4	Relevantni podaci iteracija	xxxix
2.1	Algorithm parameters.	26
2.2	Overset types field values.	29
3.1	Overset mesh system statistics.	36
3.2	Fringe assembly settings.	36
3.3	Iteration data.	37
3.4	Iteration data.	39
3.5	Iteration data.	41
3.6	NACA 4412 overset mesh system statistics.	44
3.7	Overset fringe assembly settings.	45
3.8	Iteration data for <i>backgroundMesh</i> region.	46
3.9	Iteration data for region <i>middleMesh</i>	47
3.10	Iteration data for <i>frontMesh</i> region.	49
3.11	Iteration data for <i>backgroundMesh</i> region.	53
3.12	Iteration data for <i>middleMesh</i> region.	53
3.13	Iteration data for <i>frontMesh</i> region.	53
3.14	ONR Tumblehome overset mesh system statistics.	56
3.15	Chosen iterations data.	57

4.1	Prescribed boundary conditions.	68
4.2	Hole patches.	68

List of Symbols

\tilde{A}	differential equation coefficient, [-]	8
\tilde{B}	differential equation coefficient, [-]	8
\tilde{C}	differential equation coefficient, [-]	10
\tilde{D}	differential equation coefficient, [-]	10
D	ship weight, [N]	8
A	distance from aft to CofG, [m]	70
A	distance from fore to CofG, [m]	70
N	normal reaction, [N]	8
S	suitability fraction, [-]	25
T	friction force, [N]	8
accCC	acceptor cell center position vector, [-]	34
\tilde{a}	differential equation coefficient, [-]	13
\tilde{b}	differential equation coefficient, [-]	13
dCC	donor cell center position vector, [-]	34
d	distance, [m]	70
p	differential equation coefficient, [-]	8
\mathbf{x}	spatial point	70
p	power coefficient, [-]	70
q	differential equation coefficient, [-]	8
s	ship slide along pad, [m]	8
u	interpolated value	70
w_i	weight coefficient, [-]	34

w	weight field, [-]	34
x	sway, [m]	13
y	heave, [m]	13
μ_r	dynamic friction coefficient, [-]	8
μ_r^0	static friction coefficient, [-]	8
σ	kinematic viscosity field, [kg/m^3]	74
u	interpolated value	70
φ	pad and ship inclination angle, [$^\circ$]	8
ρ	density field, [kg/m^3]	74
ψ	roll, [$^\circ$]	13
σ	surface tension coefficient, [kg/m^3]	74
θ	ship inclination angle, [$^\circ$]	8
ϑ	angle of the reaction, [$^\circ$]	10

Subscripts

a	index denoting air	74
w	index denoting water	74

Abbreviations

2D	two-dimensional	70
3D	three-dimensional	26
A.P.	aft position	65
CFD	computational fluid dynamics	ii
CofG	center of gravity	65
DCI	domain connectivity information	2
DSF	donor suitability function	3
ED	extended donor	17
F.P.	fore position	65
FVM	finite volume method	79
IDW	inverse distance weighting	70
LATUP	Launching by Turning Pads	73
MARIN	Marine Research Institute Netherlands	63
MD	master donor	17
MKV	metoda kontrolnih volumena	xxi

RDF Računalna dinamika fluida xviii

RMK računalna mehanika kontinuuma xxi

Sažetak

U ovom radu je opisana tehnika overset mreža korištena u `foam-extend` softverskom paketu, novorazvijeni algoritam za pronalazak interpolacijskog sloja, kao i teorija bočnog porinuća brodova s horizontalnog ležaja pomoću sustava zakretnih saonika. Tijekom nekoliko posljednjih desetljeća, tehnika overset mreža, poznata kao i tehnika Chimera ili preklapajućih mreža, se razvija u području računalne dinamike fluida (RDF). Tehnika overset mreža, u postupku prostorne diskretizacije računalne domene, podrazumijeva korištenje više računalnih mreža (overset regija), koje nisu povezane i generirane su zasebno. Sustav overset mreža se uobičajeno sastoji od pozadinske mreže kojom se diskretizira šire okruženje objekta, te od jedne ili više mreža kojima se prostorno diskretizira računalna domena u blizini samog objekta. Ovaj pristup najveći značaj ima u simulacijama tokova fluida u kojima je prisutno više objekata, kao i u simulacijama s pomičnim objektima i objektima u neposrednoj blizini ili zahvatu. Jednako tako se ovaj pristup koristi u parametrizacijskim i optimizacijskim proračunima. Ovaj pristup omogućuje brže i jednostavnije generiranje računalnih mreža visoke kvalitete, koje su u većini slučajeva strukturirane. Interpolacija polja fizikalnih veličina između mreža se odvija kroz tzv. interpolacijski sloj. U ovom radu je razvijen automatski algoritam za određivanje interpolacijskog sloja. Algoritam je implementiran u `foam-extend` softverskom paketu. Testiranje algoritma kao pretprocesorskog alata je provedeno na tri testna slučaja te na jednom testnom slučaju s prisustvom strujanja fluida. Prvi testni slučaj je dvo-komponentna overset mreža cilindra u kanalu. Testirana je mogućnost algoritma da pronađe interpolacijski sloj sa i bez relativnih gibanja mreža, odnosno statički i dinamički interpolacijski sloj. Drugi testni slučaj je tro-komponentna overset mreža NACA 4412 aeroprofila. Testirana je mogućnost pronalaska statičkog interpolacijskog sloja na sustavu koji se sastoji od više od dvije mreže. Treći testni slučaj je šestero-komponentna overset mreža ONR Tumblehome broda. Posljednji testni slučaj je bočno porinuće broda s horizontalnog ležaja pomoću sustava zakretnih saonika (dvo-komponentna overset mreža). Ovdje je testirana mogućnost algoritma da pronađe dinamički interpolacijski sloj u numeričkoj simulaciji s prisustvom strujanja fluida. Jednako tako, ovaj testni slučaj je korišten i za validaciju *Naval Hydro* softverskog paketa.

Ključne riječi: overset metoda, interpolacijski sloj, minimizacija preklapanja, bočno porinuće broda, `foam-extend`, *Naval Hydro* paket

Summary

In this thesis, overset mesh approach used in `foam-extend` software, newly developed fringe assembly algorithm and side launching of a ship by tipping table theory are presented. Throughout the past few decades, the overset mesh approach, also known as Chimera and overlapping mesh approach, is being developed in the field of Computational Fluid Dynamics (CFD). Overset meshing refers to the use of multiple disconnected meshes (overset regions), which are meshed separately, to discretise the flow domain. Overset mesh system typically consists of a background mesh adapted to the environment and one or more overset meshes attached to bodies, overlapping with the background mesh. This approach has much of its advantages in the computation of multiple-body and moving body problems as well as in parametric studies and optimisation analyses. The interpolation of the field data between meshes occurs through the fringe layer. For the purpose of this thesis, an automatic fringe assembly algorithm for overset meshes is developed and implemented in `foam-extend` software. Validation has been carried out on three test cases without and one test case with fluid flow present. The first test case was a two-levels mesh of a cylinder in a channel. Capability to obtain static and dynamic overset mesh assembly was tested. The second test case was a three-levels mesh of NACA 4412 airfoil, where the capability of obtaining static overset fringe assembly on overset mesh system which contains more than two meshes was tested. The third test case was six-levels ONR Tumblehome ship. The last test case was a side launching of a tanker by tipping table principle, where the capability to obtain dynamic overset fringe assembly with a presence of fluid flow, as well as *Naval Hydro Pack*, were tested.

Keywords: Overset mesh, Overset fringe assembly, Overlap minimization, Side launching of a ship, `foam-extend`, *Naval Hydro Pack*

Prošireni sažetak

Cilj ovog rada je razvoj automatskog algoritma za pronalazak receptora kod preklapajućih mreža i njegova implementacija u `foam-extend` softveru. U radu je predstavljena tehnika preklapajućih mreža, također zvana tehnikom overset ili Chimera mreža. Predstavljena je teorija bočnog porinuća brodova s horizontalnog ležaja pomoću sustava zakretnih saonika. Objašnjena je metodologija preklapajućih mreža, korištena u `foam-extend` softveru, kao i glavni pojmovi korišteni u tehnici preklapajućih mreža. Novorazvijeni algoritam je testiran na tri testna primjera kao pretprocesorski alat, te u računalnoj simulaciji bočnog porinuća broda uz prisustvo toka fluida, koja je poslužila i za validaciju *Naval Hydro* paketa koji se koristi za proračune iz područja brodogradnje i hidrodinamike u širem smislu te riječi [9].

I Uvod

Overset mreže

U Računalnoj Dinamici Fluida (RDF), Chimera tehnika, odnosno tehnika preklapajućih ili overset mreža u postupku prostorne diskretizacije računalne domene podrazumijeva korištenje više računalnih mreža (overset regija). Overset regije su generirane zasebno i nisu fizički povezane, već se sprega između regija ostvaruje preko overset interpolacijskog sloja. Sustav overset mreža se najčešće sastoji od pozadinske mreže kojom se diskretizira šire okruženje objekta, te od jedne ili više mreža kojima se prostorno diskretizira računalna domena u neposrednoj blizini samog objekta.

Tehnika preklapljenih mreža se najčešće koristi u numeričkim simulacijama u kojima je: prisutno više objekata, objekti su pomični, u neposrednoj su blizini ili su u zahvatu. Ovakav pristup omogućuje brže i jednostavnije generiranje računalnih mreža visoke kvalitete, koje su u većini slučajeva strukturirane. Također se ovaj pristup koristi u parametrizacijskim i optimizacijskim proračunima.

Teorija bočnog porinuća brodova s horizontalnog ležaja pomoću sustava zakretnih saonika

Senjanović *et al.* [1, 2, 4, 5, 6] su razvili teoriju bočnog porinuća brodova s horizontalnog ležaja pomoću sustava zakretnih saonika. Ovakav pristup omogućuje izgradnju broda na horizontalnom ležaju, kao i unaprjeđenje proizvodnoga procesa uvođenjem višeg stupnja mehanizacije i automatizacije.

Opisane su faze porinuća, koje uključuju zakretanje saonika, klizanje broda niz saonike i zakretanje broda u kolijevkama, kao i sam otplov broda. Dan je pregled sustava nelinearnih diferencijalnih jednadžbi gibanja za svaku fazu porinuća. Jednadžbe gibanja su izvedene na osnovu promatranja ravnoteže zadanih i generiranih sila, tj. težine broda i inercije, uzgona, otpora i inercije vode, te sila trenja.

II Tehnika preklapajućih mreža

U ovom poglavlju je opisana tehnika preklapajućih mreža korištena u `foam-extend` softveru. `foam-extend` je grana OpenFOAM-a, biblioteke za Računalnu Mehaniku Kontinuum (RMK) bazirane na Metodi Kontrolnih Volumena (MKV) i pisane u C++ programskom jeziku. Pojediniosti o razvojnom modelu OpenFOAM-a, kao i veza s modeliranjem i simuliranjem problema iz RMK mogu se naći u Weller *et al.* [7] i u Jasak *et al.* [8]. Opisani su najvažniji pojmovi, kao i novorazvijeni algoritam za pronalazak receptora.

Postavljanje numeričke simulacije uz korištenje preklapajućih mreža može se konceptualno podijeliti na četiri dijela, odnosno na:

- Izradu proračunskih mreža,
- Identifikaciju ćelija koje su izvan računalne domene,

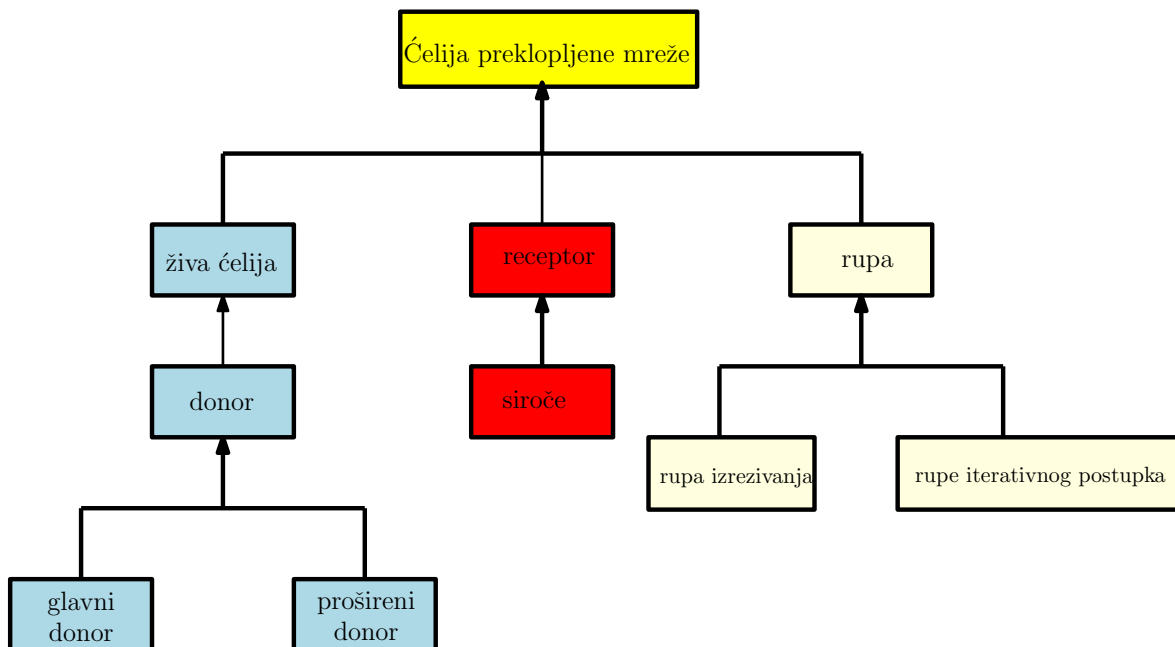
- Pronalazak interpolacijskoga sloja,
- Interpolaciju rješenja između regija.

Tipovi ćelija u tehnici preklapljenih mreža

Prema njihovoj ulozi u procesu diskretizacije jednadžbi matematičkog modela, ćelije preklapljenih mreža mogu se prikazati trima osnovnim tipovima, a to su:

- Receptor,
- Donor (aktivna ćelija) i
- Rupa (neaktivna ćelija).

Osnovni i specijalni tipovi su predstavljeni dijagramom na slici 1. Strelica pokazuje od osnovnog prema specijalnom tipu, te se shodno tome dijagram čita odozdo prema gore. Primjerice, i glavni i prošireni donor su donori, dok je donor živa ćelija, a živa ćelija je ćelija preklapljene mreže.



Slika 1: Tipovi ćelija preklapljenih mreža.

Žive ćelije služe za diskretizaciju jednadžbi matematičkog modela. Vrijednosti polja fizikalnih veličina u živim ćelijama se dobiju rješavanjem linearnoga sustava diskretiziranih jednadžbi.

Specijalni tip žive ćelije je donor. On se koristi pri interpolaciji polja između regija, tj. on je interpolacijski izvor za receptorsku ćeliju.

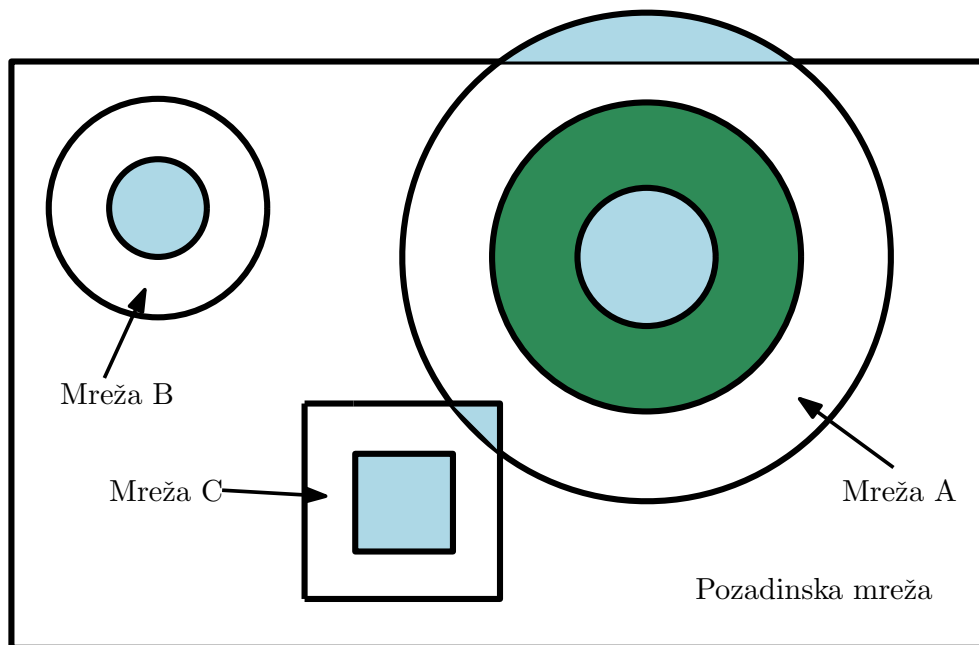
Vrijednosti polja u receptorima su dobivene isključivo interpolacijom iz donorskih ćelija. To znači da na vrijednost polja u receptorskoj ćeliji ne utječu vrijednosti polja iz susjednih ćelija, ali vrijednost polja u receptorskoj ćeliji utječe na vrijednosti polja u susjednim ćelijama. Upravo na ovaj način se ostvaruje sprega između regija između kojih je uspostavljena međumrežna komunikacija.

Prošireni donori su susjedne ćelije glavnog donora i zajedno s receptorom i glavnim donorom čine interpolacijsku molekulu. Interpolacijska molekula ovisi o odabranoj interpolacijskoj shemi.

Siroče je receptor za kojega prikladan donor nije nađen. Pojam siročeta se može poopćiti i na tzv. nevaljali donor/receptor par. Prisutnost siročadi općenito ukazuje da prilikom procesa izrade računalnih mreža nije predviđen dovoljan preklapajući sloj između mreža. Također, može ukazivati na to da preklopljene mreže nemaju sličnu rezoluciju.

Ako se mreže u blizini tijela (također zvane overset mrežama), preklapaju, potrebno je deaktivirati dijelove tih mreža koji su izvan računalne domene, tj. koji su pokriveni ostalim tijelima ili leže izvan pozadinske mreže. Proces deaktivacije dijelova mreža naziva se izrezivanje, a deaktivirane ćelije se nazivaju rupama izrezivanja. Tijekom iterativnog pronalaska receptora, pojedine ćelije se pretvaraju u rupe, koje se tada nazivaju rupama iterativnog postupka.

Na slici 2 je prikazan četvero-komponetni sustav preklopljenih mreža. Sustav se sastoji od mreža A, B, C, te od pozadinske mreže. Rubne plohe koje određuju koje ćelije će biti deaktivirane u procesu izrezivanja zovu se plohe izrezivanja. U ovom slučaju, plohe izrezivanja su vanjska ploha mreže A, kao i vanjske plohe pozadinske mreže. Rupe su obojane plavom bojom.



Slika 2: Četvero-razinski sustav preklopljenih mreža.

Tijekom iterativnoga postupka traženja interpolacijskoga sloja, receptori, koji skupa s donorima formiraju neprikladne donor/receptor parove, bivaju pretvoreni u rupe. Te rupe nazivaju se interpolacijskim rupama i na slici 2 su označene zelenom bojom.

Također je moguće da susjedne ćelije receptora budu rupe ili drugi receptori. Takve ćelije se pretvaraju u rupe tijekom postupka koji se naziva filtriranjem.

Na vrijednosti polja fizikalnih veličina u rupama ne utječu vrijednosti polja susjednih ćelija. Jednako tako, vrijednosti polja u rupama ne utječu na vrijednosti polja u susjednim ćelijama. Vrijednosti polja u rupama propisuje korisnik.

Donor/receptor parovi

Sprega između overset regija, odnosno međumrežna komunikacija se uspostavlja kroz interpolacijski sloj sastavljen od donora i receptora. Glavni donor i receptor skupa tvore donor/receptor par. Donori se traže korištenjem oktalnog stabla baziranog isključivo na dostupnim donorima. Poslije postupka izrezivanja, dostupni donori su sve ćelije koje nisu identificirane kao rupe ili receptori. Tijekom iterativnog postupka identifikacije interpolacijskog sloja, dostupne su ćelije koje prethodno nisu razmatrane kao receptori ili rupe.

Svaki receptor ima jedinstvenog donora, dok jedan donor može biti uparen s većim brojem različitih receptora. Uparivanje jednog donora s većim brojem različitih receptora može dovesti do nefizikalnog konačnog rješenja. To se događa ako preklapljene mreže nisu slične u rezolucijama. Stoga, u zoni preklapanja bi ćelije trebale biti sličnih veličina. Važno je istaknuti kako grublja od dvije preklapljene mreže određuje grešku diskretizacije.

U `foam-extend` softveru donor/receptor par je predstavljen klasom `donorAcceptor`, koja kao podatkovni član drži i logičku varijablu koja govori upada li težište pronađenog donora unutar volumena receptora. Ovaj varijabla je važan kriterij u novorazvijenom algoritmu jer je prisustvo donora koji su udaljeni od receptora nepoželjno jer može dovesti do nefizikalnog krajnjeg rješenja.

Overset regije

Kao što je prethodno rečeno, sustav overset mreža sastoji se od više mreža koje nisu fizički povezane. U `foam-extend` softveru se te mreže nazivaju overset regijama i predstavljene su klasom `oversetRegion`. Objekt klase `oversetRegion` kontrolira iterativni postupak sastavljanja interpolacijskoga sloja na način da:

1. Netko, a uobičajeno je to metoda klase `oversetMesh`, traži overset regiju receptore i donore, što pokreće lijenu evaluaciju u metodi `calcDonorAcceptorCells()`. Tijekom pronalaska interpolacijskog sloja, sve overset regije su međusobno zavisne zbog toga što se dostupnost donora/receptora mijenja tijekom iterativnoga postupka. Stoga, da bi se dobili donori i receptori iz jedne regije, pokreće se iterativni postupak u svim regijama.

2. Iterativni postupak počinje prolaskom kroz svaku regiju, tj. ažuriranjem trenutnih receptora/donora (`updateDonorAcceptors()` metoda). Postupak može biti sažet u tri koraka, a to su:
 - (a) Traženje receptora.
 - (b) Traženje donora za pronađene receptore, tj. identifikacija donor/receptor parova.
 - (c) Predavanje donor/receptor parova algoritmu za pronalazak receptora koji potom odlučuje je li prikladan interpolacijski sloj pronađen ili nije.
3. Ako je prikladan interpolacijski sloj pronađen za sve regije, donor/receptor parovi se finaliziraju pozivanjem `finaliseDonorAcceptors()` metode i alociraju se liste donora i receptora.

Prikladnost donor/receptor parova

Na ćelije koje formiraju donor/receptor parove postavlja se uvjet prikladnosti definiran po nekom od kriterija.

Automatski algoritam za pronalazak receptora kontrolira iterativni proces na način da prima donor/receptor parove, prođe kroz sve parove, provjeri jesu li prikladni po nekom od kriterija (lokalni kriterij), te u konačnici izračuna prosječnu sukladnost svih donor/receptor parova. Slijedi provjera je li globalni kriterij zadovoljen. Ako jest, iterativni postupak se završava za trenutnu regiju. U suprotnom, algoritam pronalazi nove receptore, predaje ih algoritmu koji traži donore za dane receptore, tj. formira donor/receptor parove. Novi donor/receptor parovi opet idu na provjeru u algoritmu za pronalazak receptora.

Važno je napomenuti da tzv. `calcAddressing` algoritam pronalazi inicijalne donor/receptor parove. Inicijalni receptori su ćelije čiji su susjedi rupe nastale nakon izrezivanja. Jednako tako, korisnik može definirati da se za inicijalne receptore uzmu i ćelije pojedine rubne plohe. Nakon što `calcAddressing` algoritam nađe inicijalne receptore, potrebno ih je poslati algoritmu za pronalazak donora, koji potom formirane donor/receptor parove, predaje automatskom algoritmu za pronalazak receptora.

Pronalazak prikladnog glavnog donora u automatskom algoritmu je zasnovan na funkcijama prikladnosti donora (FPD). Najčešće korištena FPD je `cellVolumes` koja računa postotak sukladnosti kao relativnu razlika obujma donora i receptora, što znači da je sukladnost S pozitivna veličina, čija vrijednost varira od 0 do 1 ($S \in \langle 0, 1 \rangle$).

Automatski algoritam za pronalazak receptora

Automatski algoritam za pronalazak receptora je kontroliran četirima parametrima i svi imaju zadanu vrijednost, koju korisnik može promijeniti. Parametri algoritma, kao i njihove zadane vrijednosti i tipovi podataka, prikazani su u tablici 1.

Tablica 1: Parametri algoritma.

Parametar	Tip podatka	Zadana vrijednost
<code>minLocalSuit</code>	<code>scalar</code>	0
<code>specifiedIterationsNumber</code>	<code>label</code>	4
<code>additionalIterations</code>	<code>Switch</code>	true
<code>orphanSuitability</code>	<code>scalar</code>	-100

Tip podatka `scalar` je ekvivalentan standardnom C++ tipu `double`, dok je `label` ekvivalentan tipu `int`. Uz parametre algoritma, korisnik treba odabrati i FPD.

Ovdje siročće ima drukčije značenje od prije spomenutoga. Siročće je receptor, odnosno donor/receptor par, za kojeg je donor pronađen, ali težište donora ne upada u volumen receptora. Prisutnost siročadi nije poželjna jer ako su ćelije koje formiraju donor/receptor par suviše udaljene jedna od druge, moguće je nefizikalno konačno rješenje.

Prikladnost siročeta se ne računa na standardan način koristeći FPD, već je ona korisnički propisana parametrom `orphanSuitability`. Propisivanjem negativne vrijednosti sukladnosti se smanjuje mogućnost da kao konačni interpolacijski sloj bude odabran onaj sloj u kojemu je siročad prisutna. Zadana vrijednost za `orphanSuitability` je -100% što znači da ćelije koje formiraju donor/receptor par nisu uopće prikladne.

Nakon što `calcAddressing` nađe inicijalne donor/receptor parove, kreće automatski algoritam za pronalazak receptora. Algoritam prolazi kroz sve donor/receptor parove, te provjerava je li trenutni par siroče ili nije. Ako jest, propisuje mu prikladnost i sprema ga u listu neprikladnih parova. U suprotnom računa prikladnost koristeći FPD, te trenutni donor/receptor par kategorizira kao prikladan ili kao neprikladan na osnovu korisnički propisanog kriterija u FPD-u. Prikladne parove sprema u listu prikladnih parova koju drži tijekom cijeloga iterativnoga procesa, dok neprikladne parove sprema u listu neprikladnih parova.

Nakon što algoritam prođe kroz sve parove, računa prosječnu prikladnost i sve parove sprema u listu za pohranu. Ovdje je važno naglasiti da u listu za pohranu idu i parovi koji su identificirani kao prikladni iz prethodnih iteracija.

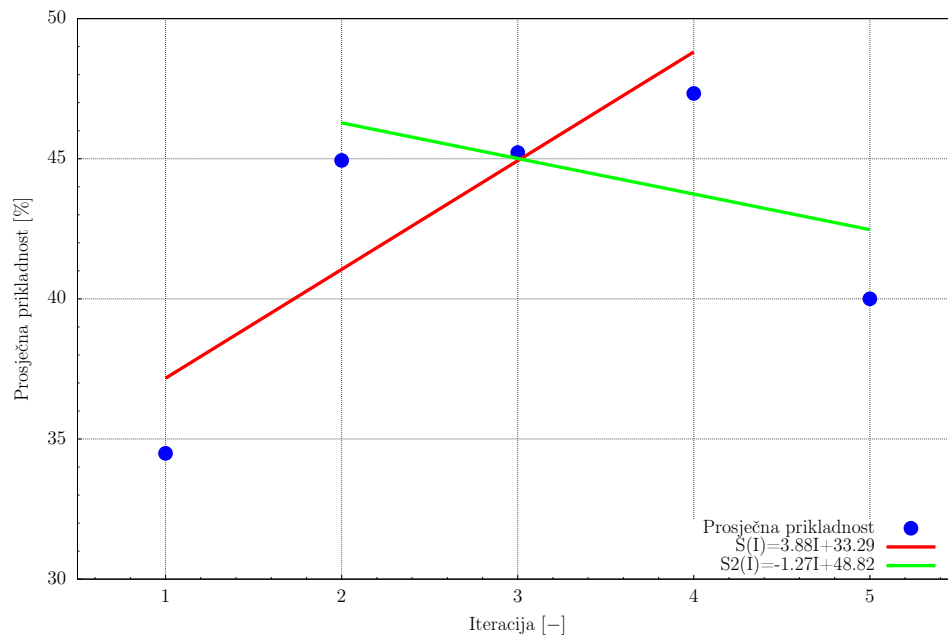
Algoritam prati povijest iterativnoga postupka na način da nakon svake napravljene iteracije instancira objekt klase `iterationData`. Klasa `iterationData` kao podatkovne članove drži sve što je potrebno za rekonstrukciju interpolacijskog sloja i računanje koeficijenta linearne regresije, a to je:

- Lista donor/receptor parova (prethodno spomenuta lista za pohranu),
- Lista rupa,
- Prosječna prikladnost svih parova i
- Broj iteracije.

Objekti klase `iterationData` se spremaju u `FIFOStack` (eng. First In First Out) kontejner. Nakon što je korisnički definiran broj iteracija napravljen (parametar `specifiedIterationsNumber` i ako je korisnik definirao da želi da se dodatne iteracije rade (parametar `additionalIterations`), algoritam računa nagib pravca, odnosno gradijent poklapanja koristeći linearnu regresiju. Ovdje je redni broj iteracije regresor, dok je proječna prikladnost zavisna varijabla.

Polazi se od pretpostavke da ako je gradijent prikladnosti pozitivan, u sljedećim iteracijama je moguće naći interpolacijski sloj čije je poklapanje više, pa tako algoritam radi još jednu dodatnu iteraciju. Prvi element kontejnera se izbacuje, pa je na taj način u kontejneru uvijek prisutan `specifiedIterationsNumber` ili `specifiedIterationsNumber-1` broj elemenata.

Na slici 3 prikazana je međuovisnost broja iteracije o prosječnoj prikladnosti. U ovom slučaju, nakon što su četiri iteracije napravljene, izračunat je gradijent prikladnosti. Zbog pozitivnog gradijenta prikladnosti, prvi element kontejnera, tj. iteracija 1, je izbačena, nakon čega se u kontejneru nalaze tri elementa. Napravljena je jedna dodatna iteracija, tj. iteracija 5. Nakon što je peta iteracija napravljena, zbog negativnog gradijenta prikladnosti, iterativni proces je završen. Za konačni sastavljanje konačnog interpolacijskog sloja, izabrana je iteracija u kojoj je bila najviša prosječna prikladnost, tj. iteracija 4.



Slika 3: Dijagram iterativnog procesa.

Algoritam provjerava postoji li u trenutnoj iteraciji neprikladnih donor/receptor parova i je li željeni broj iteracija napravljen. Ova dva uvjeta definiraju globalni kriterij. Ako globalni kriterij nije zadovoljen, pronalaze se novi receptori. Novi receptori su neposredni susjedi receptora koji formiraju neprikladne donor/receptor parove. Receptori moraju zadovoljiti kriterij dostupnosti - dostupni su samo ako nisu rupe ili ako nisu razmatrani kao receptori u prethodnim iteracijama. Algoritam filtriranja pretvara receptore iz neprikladnih donor/receptor parova u rupe, tj. stavlja ih u listu rupa.

U suprotnom, tj. ako je globalni kriterij zadovoljen, iterativni postupak je završen za trenutnu regiju. Dakle, prema zadanim postavkama iterativni postupak je gotov ako:

- Korisnički propisani kriteriji su zadovoljeni, odnosno u trenutnoj iteraciji više nema neprikladnih donor/receptor parova ili,
- Ako je specificirani broj iteracija napravljen i gradijent prikladnosti je negativan ili,
- Ako je dodatna iteracija napravljena, a gradijent poklapanja je negativan.

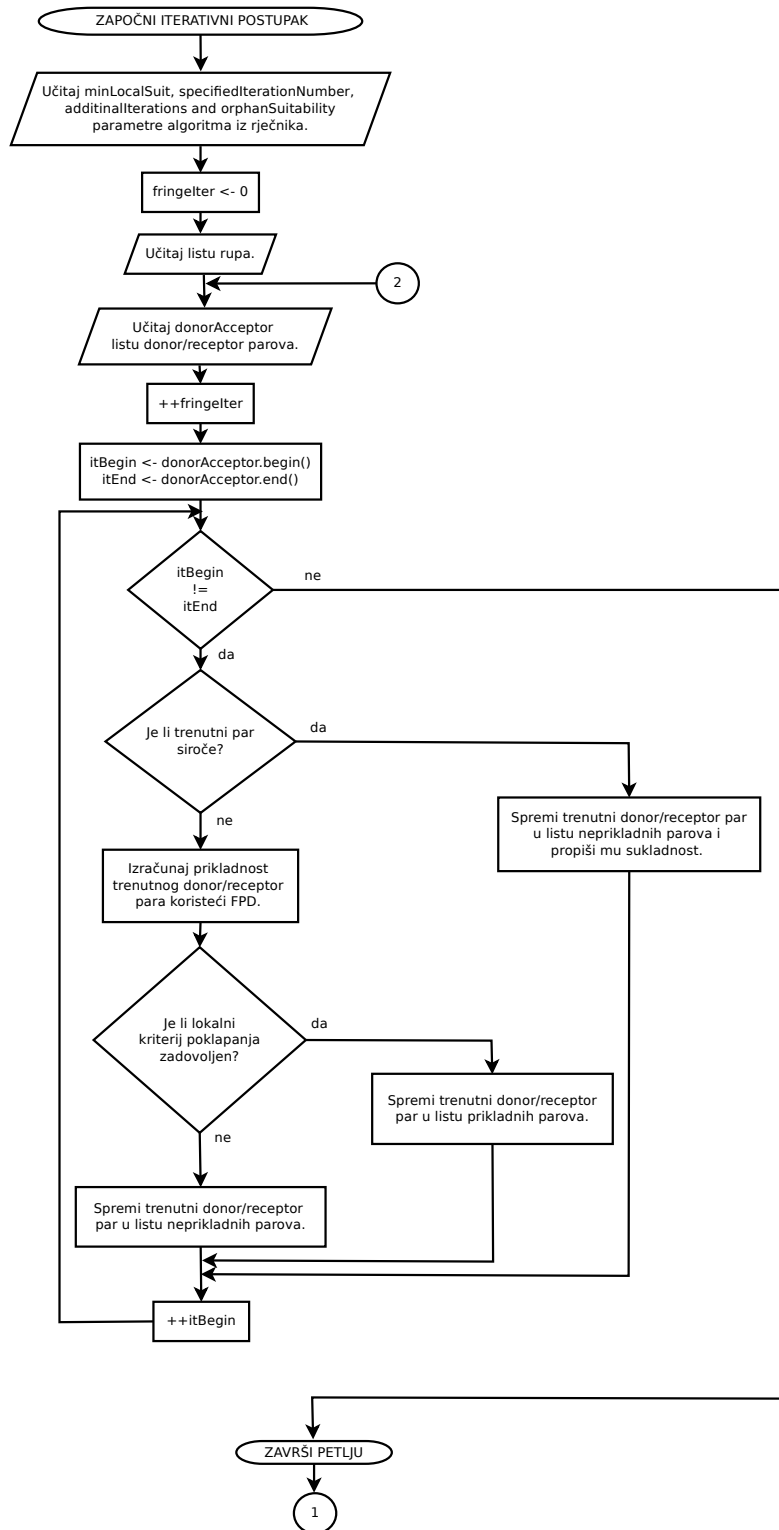
Nakon što je iterativni postupak gotov, algoritam prolazi kroz sve objekte iz `iterationDataHistory` kontejnera i pronalazi onoga s najvišom prosječnom sukladnošću.

Posljedni korak je filtracija donor/receptor parova. Moguće je da pojedini receptor u potpunosti bude okružen drugim receptorima ili rupama, pa se takvi receptori pretvaraju u rupe tijekom filtracijskog algoritma. Dijagram toka algoritma prikazan je na slikama [4](#) i [5](#).

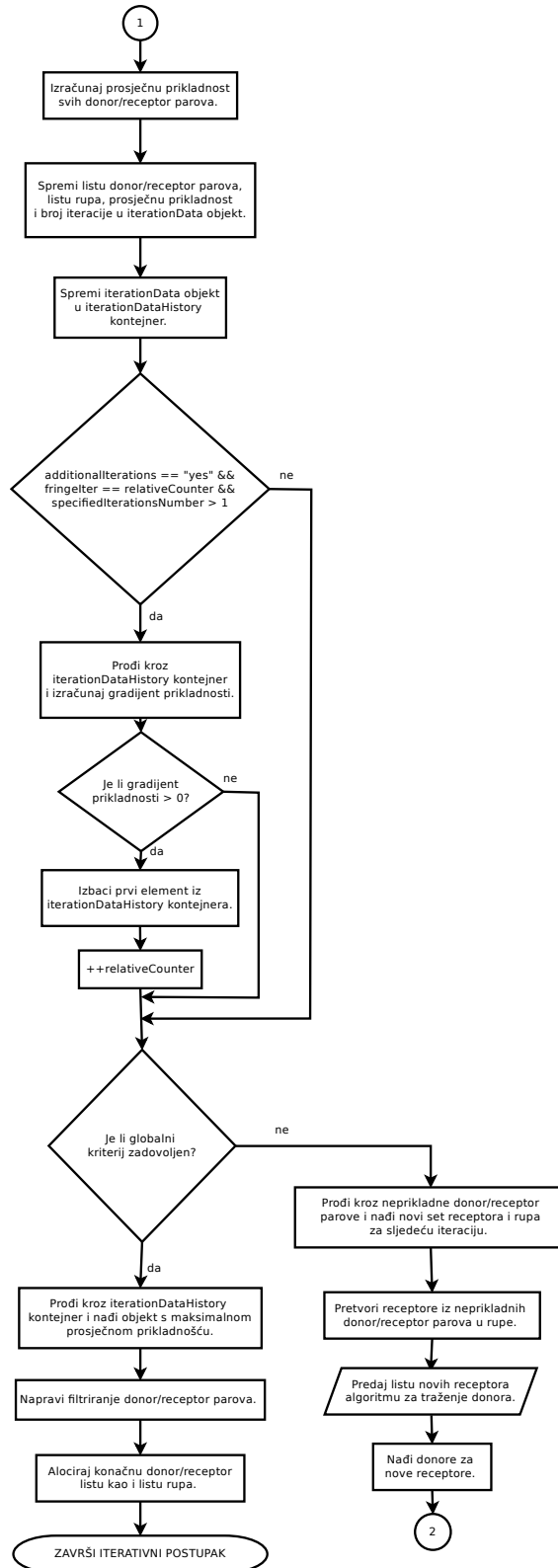
Nakon što su tipovi ćelija identificirani, inicijalizira se volumno skalarno polje tipova ćelija. Inicijalna vrijednost je 0, što predstavlja živu ćeliju. Vrijednosti polja za ostale tipove ćelija su prikazane u tablici [2](#).

Tablica 2: Vrijednosti volumnoga polja tipova ćelija.

Tip ćelije	Vrijednosti polja
receptor	2
donor	1
živa ćelija	0
rupa	-1



Slika 4: Dijagram toka algoritma (prvi dio).



Slika 5: Dijagram toka algoritma (drugi dio).

Interpolacija polja fizikalnih veličina između preklopljenih mreža

Interpolacija između regija može biti implicitna ili eksplicitna. Implicitna interpolacija pruža snažnu spregu, dok eksplicitna pruža slabu spregu između regija.

Interpolacija polja fizikalnih veličina se odvija unutar računalne molekule. Izgled računalne molekule ovisan je o interpolacijskoj shemi.

Kod injekcijske interpolacijske sheme, računalnu molekulu čine receptor i glavni donor. Vrijednost polja fizikalne veličine se iz donora injektira u receptor.

Kod interpolacijske sheme prosječnih vrijednosti, računalnu molekulu čine receptor, glavni i prošireni donori. Interpolirana je vrijednost srednja vrijednost polja donora i proširenih donora.

U ovom radu je korištena shema inverzne udaljenosti. Interpolacijska molekula je sačinjena kao i u interpolacijskoj shemi prosječnih vrijednosti. Ovdje se za izračun interpolirane vrijednosti koriste težinski faktori, na način da vrijednost polja u donoru koji je udaljeniji od receptora ima manji utjecaj na interpoliranu vrijednost.

Da bi se uspostavila interpolacija između regija, vanjskoj rubnoj plohi mreže oko tijela (overset regije) se propisuje **overset** rubni uvjet.

III Testiranje algoritma

U ovom poglavlju je novorazvijeni algoritam testiran kao pretprocesorski alat na tri testna primjera.

Prvi testni primjer je dvo-komponentna sustav preklopljenih mreža cilindra u kanalu. Ovdje je testirana mogućnost sastavljanja kako statičkoga, tako i dinamičkoga interpolacijskoga sloja.

Drugi testni primjer je tro-komponentni sustav preklopljenih mreža NACA 4412 aeroprofila. Testirana je mogućnost sastavljanja statičkoga interpolacijskoga sloja na sustavu koji je sastavljen od više od dvije regije.

Treći testni primjer je šestero-komponentni sustav ONR Tumblehome broda. Testirana je mogućnost pronalaska statičkoga interpolacijskoga sloja na trodimenzionalnim, nestrukturiranim proračunskim mrežama.

Poradi svoje jednostavnosti i lakoće demonstracije koraka algoritma, ovdje je prikazan sustav cilindra u kanalu. Ovaj testni primjer je važan test kriterija zaustavljanja, kao i odabira najprikladnijega interpolacijskog sloja.

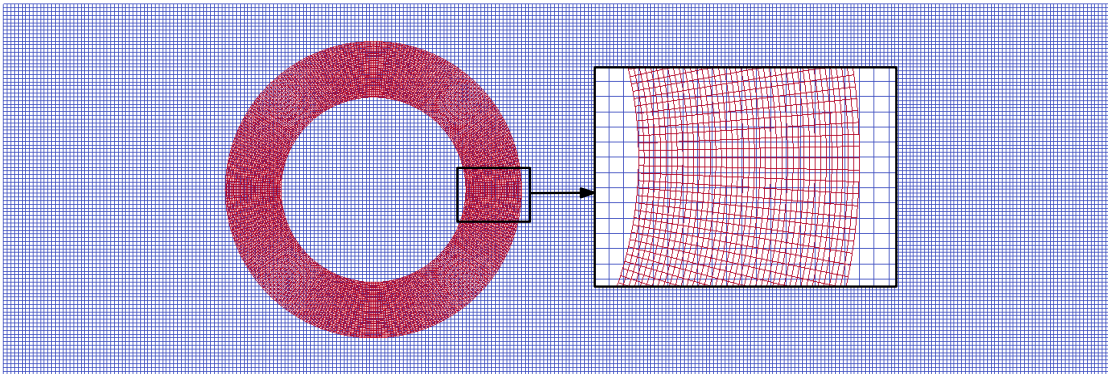
Dvo-komponentni sustav preklapljenih mreža cilindra u kanalu

Sustav preklapljenih mreža sastoji se od dvije overset regije. Prva regija je pozadinska mreža, dok je druga regija mreža oko cilindra. Obje mreže su dvodimenzionalne. Pozadinska mreža je prostorno diskretizirana jednakim ćelijama. Na taj način rast ili pad prikladnosti donor/receptor parova ovisi isključivo o prostornoj diskretizaciji mreže oko cilindra.

Test 1 (standardna gradacija)

Na slici 6 prikazan je sustav mreža u kojemu je overset regija diskretizirana na standardan način. Kako bi se opisala fizika graničnoga sloja, najsitnije ćelije se nalaze uz sam rub cilindra. Širina ćelija, a sukladno tome njihova duljina i obujam, raste s povećanjem radijalne koordinate cilindra, odnosno, najveće ćelije se nalaze na vanjskom rubu overset mreže.

Za ovaj testni primjer, korišten je novorazvijeni algoritam s propisanom maksimalnom vrijednošću za minimalno lokalno poklapanje (`minLocalSuit = 100%`). Korištena je `cellVolumes` FPD. Ostale vrijednosti parametara algoritma nisu mijenjane u odnosu na zadane. Zid cilindra je definiran kao rubna ploha koja određuje koje ćelije će biti deaktivirane.



Slika 6: Pozadinska i overset mreža s detaljnim pogledom (standardna gradacija).

U tablici 3 dani su podaci vezani za pojedine iteracije. Dan je broj iteracije, broj neprikladnih parova, prosječna prikladnost i gradijent poklapanja. Broj neprikladnih parova se dijeli na broj siročadi i na broj parova koji ne zadovoljavaju lokalni kriterij prikladnosti.

Iz tablice 3 je vidljivo kako su svi donor/receptor parovi kategorizirani kao neprikladni. To se događa zbog toga što je propisan nerealno velik lokalni kriterij sukladnosti. Inicijalni donor/receptor parovi su oni odmah uz unutarnji rub cilindra. Algoritam kategorizira sve parove kao neprikladne, pa stoga za nove receptore uzima sljedeći sloj ćelija, odnosno širi se prema vanjskom rubu cilindra. Vidljivo je da prosječna prikladnost parova kontinuirano raste tijekom iterativnoga postupka. Jednako tako, gradijent prikladnosti je stalno pozitivan. To se događa zato što su ćelije pozadinske regije veće po kriteriju obujma od ćelija overset regije, a algoritam propagira u smjeru povećanja obujma.

Tablica 3: Relevantni podaci iteracija.

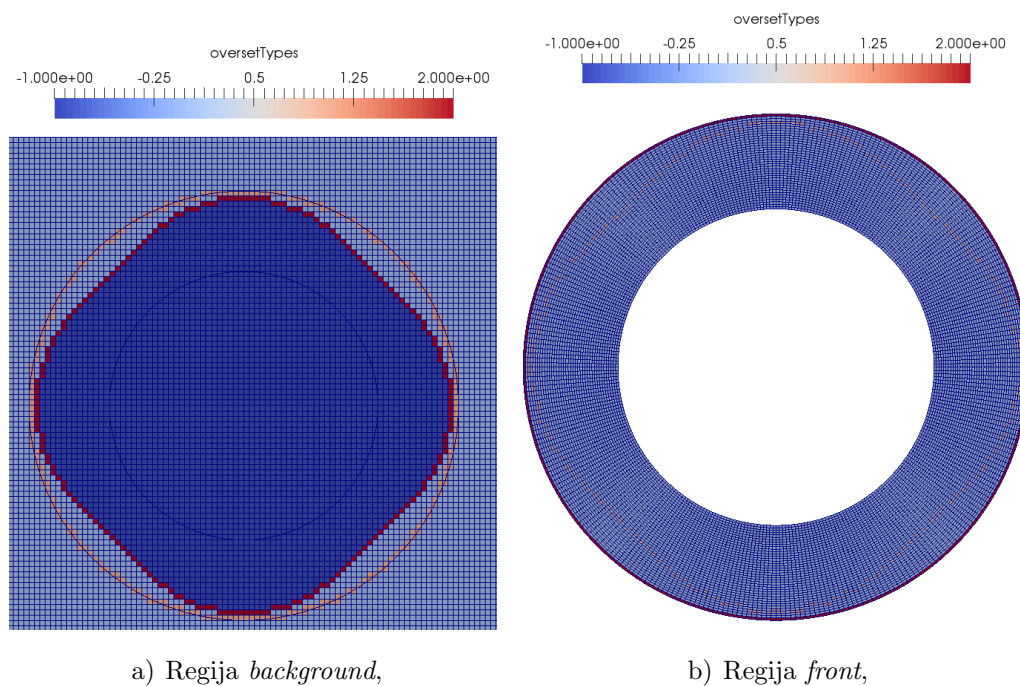
Iteracija	Broj neprikladnih parova	Prosječna prikladnost [%]	Gradijent poklapanja [%/iter]
1	0 / 144	18.76	/
2	0 / 148	19.45	/
3	0 / 152	20.08	/
4	0 / 156	20.77	0.66
5	0 / 160	21.41	0.65
6	0 / 164	22.05	0.65
7	0 / 168	22.73	0.65
8	0 / 172	23.36	0.65
9	0 / 176	24.01	0.65
10	0 / 180	24.67	0.64
11	0 / 184	25.32	0.65
12	0 / 188	25.94	0.64
13	0 / 192	26.61	0.64
14	0 / 196	27.21	0.63
15	12 / 188	20.10	-1.69

Nakon što su prve četiri iteracije napravljene, izračunat je gradijent poklapanja. Budući da je njegova vrijednost pozitivan broj, prva iteracija je odbačena i napravljena je još jedna iteracija. Opet je izračunat gradijent poklapanja, koji je opet bio pozitivan i napravljena je dodatna iteracija.

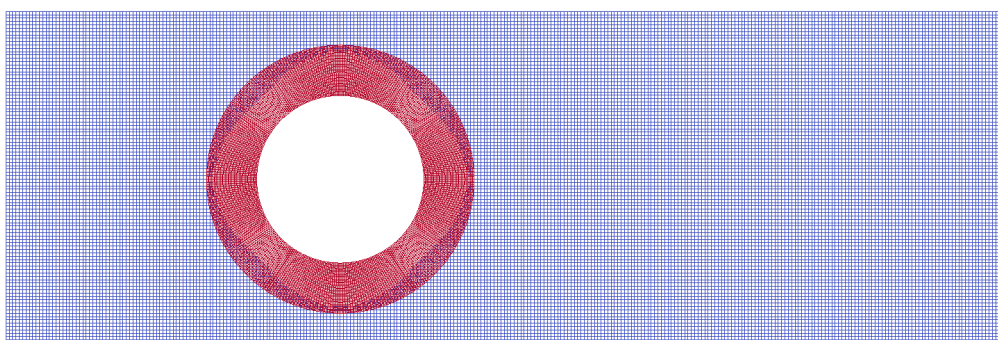
Budući da vrijednost poklapanja kontinuirano raste, iterativni proces propagira prema vanjskom rubu cilindra. U 15.-oj iteraciji se pojavljuje 12 siročadi. To se događa zato što je algoritam došao do kraja overset regije i nema više dostupnih receptora, već bi za nove receptore trebao uzeti donore za pozadinsku mrežu. Ovdje je prisustvo siročadi degradiralo prosječnu prikladnost, što je rezultiralo negativnim gradijentom poklapanja.

Uslijed negativnoga gradijenta poklapanja, iterativni proces se završio. Između četiri iteracije, odnosno između iteracija pod brojem 12, 13, 14 i 15 algoritam bira onu s najvećom prosječnom prikladnošću kako bi sastavio interpolacijski sloj, a to je 14. iteracija.

Na slici 7 prikazana su polja tipova overset ćelija za pojedine regije, dok je na slici 8 prikazan konačni sustav preklopljenih mreža.



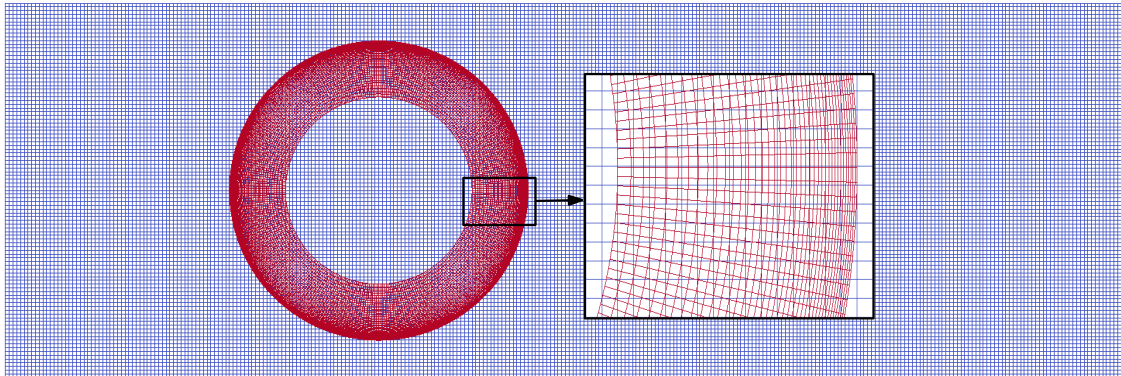
Slika 7: Polja tipova overset ćelija.



Slika 8: Konačni sustav preklopljenih mreža.

Test 2 (obrnuta gradacija)

Na slici 9 prikazan je sustav mreža u kojemu je overset regija diskretizirana s obratnom gradacijom od uobičajene. Tako se najsitnije ćelije nalaze na vanjskom dijelu overset regije.



Slika 9: Pozadinska i overset mreža s detaljnim pogledom (obratna gradacija).

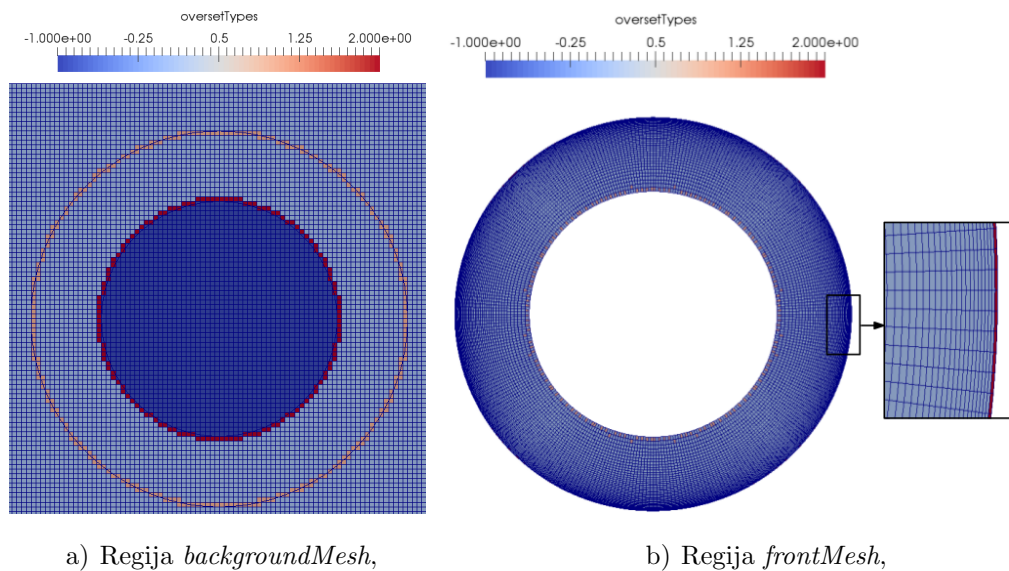
Algoritam je testiran s jednakim postavkama kao i u prethodnom testnom primjeru. U tablici 4 dani su relevantni podaci iteracija vezani za ovaj testni slučaj. Inicijalni sloj receptora se nalazio uz sam zid cilindra, gdje se nalaze najveće ćelije. Budući da je propisana nerealno visoka vrijednost minimalne lokalne prikladnosti, algoritam svaki set receptora kategorizira neprikladnim, te na taj način propagira sloj po sloj ćelija šireći se prema vanjskoj rubnoj plohi cilindra. Na taj način nailazi na sve sitnije ćelije po kriteriju obujma, što ukazuje na to da se srednja vrijednost prikladnosti kontinuirano smanjuje.

Nakon što je napravljen propisan broj iteracija, izračunat je gradijent prikladnosti. Budući da je gradijent prikladnosti negativan, algoritam između iteracija 1, 2, 3 i 4 bira onu s najvećom prosječnom prikladnosti, a to je iteracija broj 1.

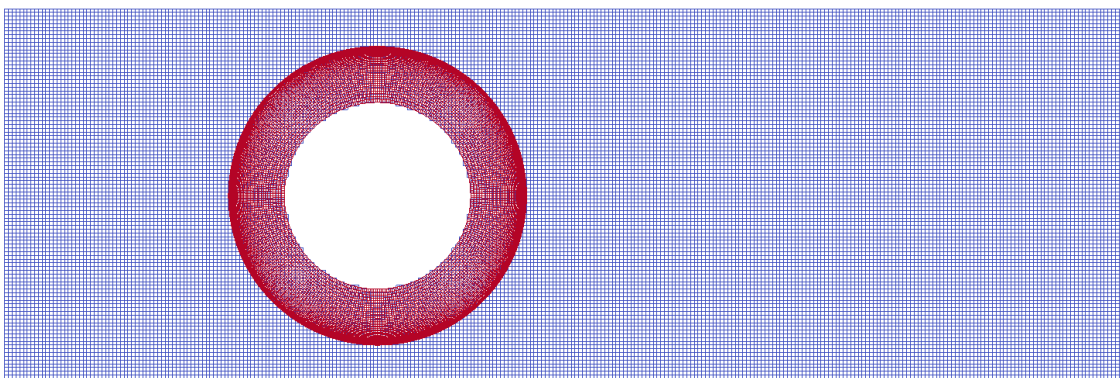
Na slici 10 prikazana su polja tipova overset ćelija za pojedine regije, dok je na slici 11 prikazan konačni sustav preklapljenih mreža.

Tablica 4: Relevantni podaci iteracija

Iteracija	Broj neprikladnih parova	Prosječna prikladnost [%]	Gradijent prikladnosti [%/iter]
1	0 / 144	41.36	/
2	0 / 148	40.5	/
3	0 / 152	39.52	/
4	0 / 156	38.43	-0.97



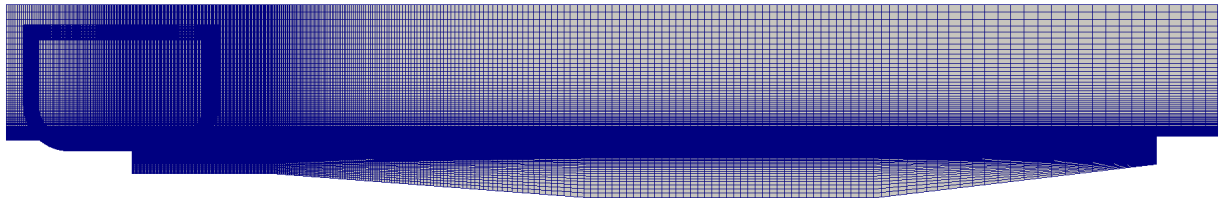
Slika 10: Polja tipova overset ćelija.



Slika 11: Konačni sustav preklapljenih mreža.

IV Numerička simulacija bočnoga porinuća tankera od 260 000 DWT s horizontalnog ležaja pomoću sustava zakretnih saonika

Ova simulacija je provedena u svrhu testiranja novorazvijenog algoritma kao i validacije *Naval Hydro* paketa. Sustav preklapljenih mreža sastoji se od dvije regije, pozadinske regije i overset regije, koja predstavlja mrežu oko tankera. Modelska ispitivanja su provedena u nizozemskom brogarskom institutu pod nazivom *Maritime Research Institute Netherlands* (MARIN). Bazen u kojem su se vršila ispitivanja napravljen je prema topologiji dna splitskoga brodogradilišta "Brodosplit". Skladno danoj topologiji dna, napravljena je i računalna domena. Prostorno diskretizirana računalna domena prikazana je na slici 12.



Slika 12: Prikaz sustava preklapljenih mreža u početnom trenutku

Proveden je dvodimenzionalni proračun bočnoga porinuća broda s propisanim gibanjima. Propisana su gibanja mjerena tijekom modelskog ispitivanja (gibanja su metodom sličnosti prevedena na stvarnu skalu), kao i gibanja koja su dobivena kao rješenje nelinearnog sustava jednadžbi gibanja.

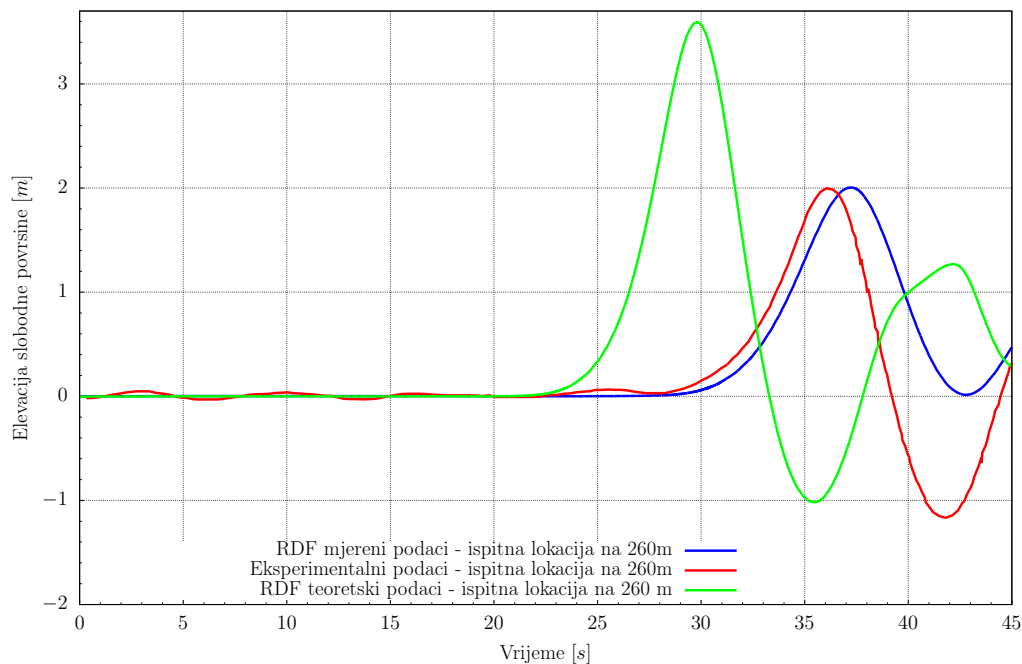
Kao referentna veličina za validaciju, uzeta je amplituda prvog vala. Amplituda prvog vala je od interesa jer je val generiran prodorom broda u bazen potopio desnu obalu bazena, a reflektirani val lijevu.

Eksperimentalna gibanja su digitalizirana iz dostupnih dijagrama. Mjerena su gibanja krme i pramca, pa su horizontalna i vertikalna gibanja za potrebe dvodimenzionalnog proračuna interpolirana na težište broda. Digitalizacijom i interpolacijom propisanih gibanja unešena je greška u proračun. Budući da su eksperimentalna i teoretska gibanja dostupna samo za 28 s, kako bi se uhvatila visina prvog vala, gibanja su ekstrapolirana, pa se uzdizanje slobodne površine iza tog vremenskog trenutka ne uzima kao relevantno.

Na slici 13 je prikazano uzdizanje slobodne površine na lokaciji mjerenja. Dani su rezultati modelskih mjerenja, kao i rezultati numeričkih simulacija s propisanim eksperimentalnim i teoretskim gibanjima.

Maksimalna izmjerena valna amplituda je 1.997 m i izmjerena je u vremenskom trenutku $t = 36.04$ s. Maksimalna izračunata valna amplituda iz simulacije s propisanim eksperimentalnim gibanjima je 2.004 m u vremenskom trenutku $t = 37.22$ s. Relativna greška za valnu amplitudu je 0.36%, dok je relativna greška za pomak u fazi 3.16%. Ovi rezultati pokazuju jako dobro poklapanje s eksperimentalnima.

Izračunata amplituda drugog vala je manja od mjerene. To se događa zbog prirode dvodimenzionalnog računalnog proračuna. Prednja i stražnja strana računalne domene ne dozvoljavaju protok fluida. Kada se tanker porine, njegov ulazak u "numerički bazen" podiže slobodnu površinu za otprilike 0.5 m. Navedeni poremećaj propagira s prvim valom, tako da samo prva valna amplituda treba biti uzeta kao relevantna.



Slika 13: Uzdizanje slobodne površine na lokaciji mjerenja

Numerički rezultati dobiveni s teoretskim gibanjima daju ponešto veća odstupanja u valnoj amplitudi i fazi. To se događa zato što težište broda, po teoretskom proračunu, potone dublje nego što je slučaj u stvarnosti. Teoretski proračun je stoga na strani sigurnosti što se tiče potrebne dubine dna brodogradilišta u kojemu će se odvijati bočno porinuće.

V Zaključak

U ovom radu je proveden razvoj, implementacija i testiranje automatskog algoritma za pronalazak receptora kod preklapajućih mreža. Algoritam je implementiran u `foam-extend` softveru otvorenog koda, specijaliziran za numeričke proračune iz područja računalne mehanike kontinuuma [7, 8]. Softver je baziran na metodi kontrolnih volumena. Jednako tako, provedena je i validacija *Naval Hydro* paketa.

Novorazvijeni algoritam određuje interpolacijski sloj robusnije i pouzdanije. Algoritam prati povijest iteracija i za konačni interpolacijski sloj odabire iteraciju u kojoj je prosječna prikladnost najviša, tj. ima implementiran povratni mehanizam. Algoritam koristi modificirane FPD koje pružaju realističan uvid u prikladnost ćelija. Zavisno o gradijentu poklapanja, algoritam radi dodatnu iteraciju ili zaustavlja iterativni proces za trenutnu regiju. Prikladnost siročadi se ne računa na standardan način koristeći FPD, već se propisuje negativna vrijednost. Propisivanjem negativne vrijednosti, degradira se prosječna prikladnost. Na taj način se smanjuje mogućnost da se za konačni interpolacijski sloj odabere iteracija u kojoj je siroče prisutno.

Kriterij zaustavljanja je bolje definiran. Kada algoritam propagira sve do ruba trenutne regije, nedostatak raspoloživih ćelija, tj. prisutnost siročadi, degradira prosječnu prikladnost, što u konačnici rezultira negativnim gradijentom prikladnosti i zaustavljanjem iterativnog procesa za trenutnu regiju.

Algoritam je, kao pretprocesorski alat, testiran na tri testna primjera, te na jednom testnom primjeru s prisutnošću toka fluida. Pretprocesorski testni primjeri su: dvo-komponentni sustav preklapljenih mreža cilindra u kanalu, tro-komponentni sustav NACA 4412 aeroprofila, te šestero-komponentni sustav ONR Tumblehome broda. Parametri algoritma su varirani, kao i gradacija mreže oko cilindra.

Pretprocesorski testovi pokazuju da algoritam producira valjane statičke i dinamičke interpolacijske slojeve, čak i s nerazumno visoko propisanim kriterijom za minimalno lokalno poklapanje. Pokazano je kako je propisana prikladnost siročeta od samo -7% bila dovoljna da zaustavi iterativni postupak u testnom primjeru cilindra u kanalu sa standardnom gradacijom. Primijećene su nepoželjne rupe u testnom primjeru ONR Tumblehome broda, kao rezultat pogreške u proceduri izrezivanja. Detaljna analiza ONR Tumblehome testnog primjera treba biti provedena u budućem radu.

Zadnji testni primjer je numerička simulacija bočnog porinuća tankera od 260 000 DWT s horizontalnoga ležaja pomoću sustava zakretnih saonika. U usporedbi s eksperimentalnim rezultatima, primijećena je neznatna razlika u valnoj amplitudi prvog vala, kao i neznatan pomak u fazi prvog vala. Relativna pogreška za valnu amplitudu je 0.36%, dok je za pomak u fazi 3.16%. Rezultati ovoga testnog slučaja demonstriraju valjanost novorazvijenoga algoritma, kao i *Naval Hydro* paketa.

Zamijećena je primijetna razlika između RDF rezultata s propisanim eksperimentalnim gibanjima i RDF rezultata s propisanim teoretskim gibanjima. To se događa zato što težište broda, po teoretskom proračunu, potone dublje nego što je slučaj u stvarnosti. Stoga je zaključeno da je elevacija slobodne površine jako osjetljiva na propisana gibanja.

Također, zaključuje se da je 2D proračun prikladan alat samo za izračun prve valne amplitude zbog prirode 2D proračuna u kojem rubni uvjeti ne dopuštaju tok fluida kroz prednju i stražnju stranu računalne domene, pa se slobodna površina podiže za otprilike 0.5 m. Ovaj poremećaj se širi s prvim valom. Također se javljaju trodimenzionalni efekti difrakcija i radijacija.

1 Introduction

In this chapter, the overset mesh technique, also called Chimera and overlapping mesh technique, is introduced. An overview of automatic fringe assembly algorithms as used in different commercial and scientific codes is given. The theory of side launching of a ship on an even keel by turning pads on circular slipways as a two-dimensional problem, originally developed by Senjanović *et al.* [1, 2, 4, 5, 6], is introduced and briefly described.

1.1. Overset Mesh

In Computational Fluid Dynamics (CFD) overset meshing refers to the use of multiple disconnected meshes to discretize the flow domain. The component meshes, which can be any size, type, or shape, need only overlap each other to cover the solution domain completely [10]. Furthermore, a component mesh resolving one geometric feature may intersect another geometric feature [12].

Mesh components can be added or altered to represent the arbitrary shape of real-life geometries, thus permitting a great deal of flexibility in the discretization of multi-connected domains. This approach has much of its advantages in the computation of multiple-body and moving body problems as well as in optimization-study problems [12].

The power inherent in the simple concept of disconnecting domain connectivity from mesh construction cannot be overstated. In addition to simplifying the mesh generation, component meshes can now be tailored to the local geometry, physics, and even solution

model, e.g. turbulent flow transition or ice accretion. Compromises in mesh quality needed to facilitate domain connectivity have been shown to reduce simulation accuracy and robustness. By using overset mesh technology, such problems can be mitigated [10].

Overset mesh generation is then conceptually split into off-body or background meshes and near-body meshes which resolve geometry and boundary layers. Structured hexahedral component meshes are often used for their efficiency and accuracy. However, the overset technique is routinely applied using hybrid unstructured meshes for highly automated meshing of complex configurations [10].

The overset approach also enables changing the geometry and mesh system locally without requiring regeneration of other meshes. This flexibility greatly simplifies design studies as geometry perturbations can easily be added to an existing design and mesh system by meshing the new feature and possibly including meshes to connect the new feature with the existing meshes. Since the baseline mesh system is not altered the changes in the flow are more reflective of the change in the geometry and not changes resulting from remeshing the entire geometry [11].

The use of an overset mesh system is also an enabling technology for the simulation of bodies in relative motion where geometry components or whole bodies move relative to one another. This capability has been widely used for aircraft such as weapon separation, where a bomb or missile is dropped from a parent aircraft, and rotorcraft for high fidelity simulations of helicopters with blades that may rotate, flap, and flex relative to the fuselage. Candidate hydrodynamic applications include ships or submarines with rotating propulsors, the launch of torpedo or mini-submarines from a parent ship, moving control surfaces, ship motion relative to the sea surface, and sea keeping simulations with multiple ships in close proximity [11].

A CFD solution on the system of meshes requires coupling the solution between meshes in the overlapped regions. This is typically performed by identifying appropriate intermesh boundary locations in one mesh and obtaining the value to be applied by interpolating the solution from meshes that overlap the region [11].

The Domain Connectivity Information (DCI) consists of the locations that are to be excluded from the computation, the location of the intermesh boundary locations, and the corresponding interpolation sources. This domain connectivity information is computed by a code typically called an overset mesh fringe assembly code [11]. The goal of this thesis is to develop an automatic fringe assembly algorithm for overset meshes,

which includes finding the most suitable set of acceptors in a reliable way.

1.2. An Overview of Overset Mesh Codes

Sugar++ is a general purpose overset mesh assembly code. It can be run as a pre-processor for static configurations or linked into the flow solver as a library to perform the overset DCI computations at each step of a time-dependent configuration [13]. It provides overlap minimization, i.e. fringe assembly using general Donor Suitability Function (DSF). Further information about Sugar++ and its fringe assembly algorithm can be found in the presentation about Sugar++ capabilities [13].

STAR-CCM+ is a commercial software for computational continuum mechanics which has implemented overset mesh technology. After the overset mesh is created, the hole-cutting process automatically couples the overset region with the background region through the overset interface. Fringe assembly can be obtained by layer or by global approach. Further information about fringe assembly procedure may be found in [14].

PEGASUS 5 is a code which performs a pre-processing step for the Overset CFD method developed by NASA. The code prepares the overset volume meshes for the flow solver by computing the domain connectivity database, and blanking out mesh points which are contained inside a solid body. Further information about PEGASUS 5 code may be found at official website [15].

ANSYS Fluent software is the tool for CFD which has overset mesh capability. By default, smaller cells have a higher donor priority. Thus, in mesh overlapping areas, the solver attempts to obtain the solution on the finest local mesh. The resulting mesh interface moves to an area where the meshes are more comparable in cell size, leading to better solution quality [16].

1.3. Launching Theory of Ships from Horizontal Berth by Tipping Table Arrangement

In this chapter, the theory of side launching of a tanker on an even keel by turning pads on circular slipways as a two-dimensional problem, originally developed by

Senjanović *et al.* [1, 2, 4, 5, 6], is introduced and briefly described.

Launching of ships by a tipping table arrangement makes their building on a horizontal berth possible. This advantage enables the improvement of the production process by applying a higher degree of mechanisation and automation [1].

The launching system includes a set of concrete circular slipways, which are built-in within the quay, and the steel turning pads. The sliding surface of the pads is an extension of the sliding lines of the building berth. Launching system is presented in Figure 1.1.

A structure assembled on the wood packing is pushed onto the braked tipping table by hydraulic gripper jacks. The tension forces remove the triggers when the moment of structure weight overcomes the friction moment on the sliding arches. As a result, the rotation of the structure with the pads begins and lasts until the buoyancy excites the sliding of the structures along the pads, and then until its erection in cradles, up to floating.

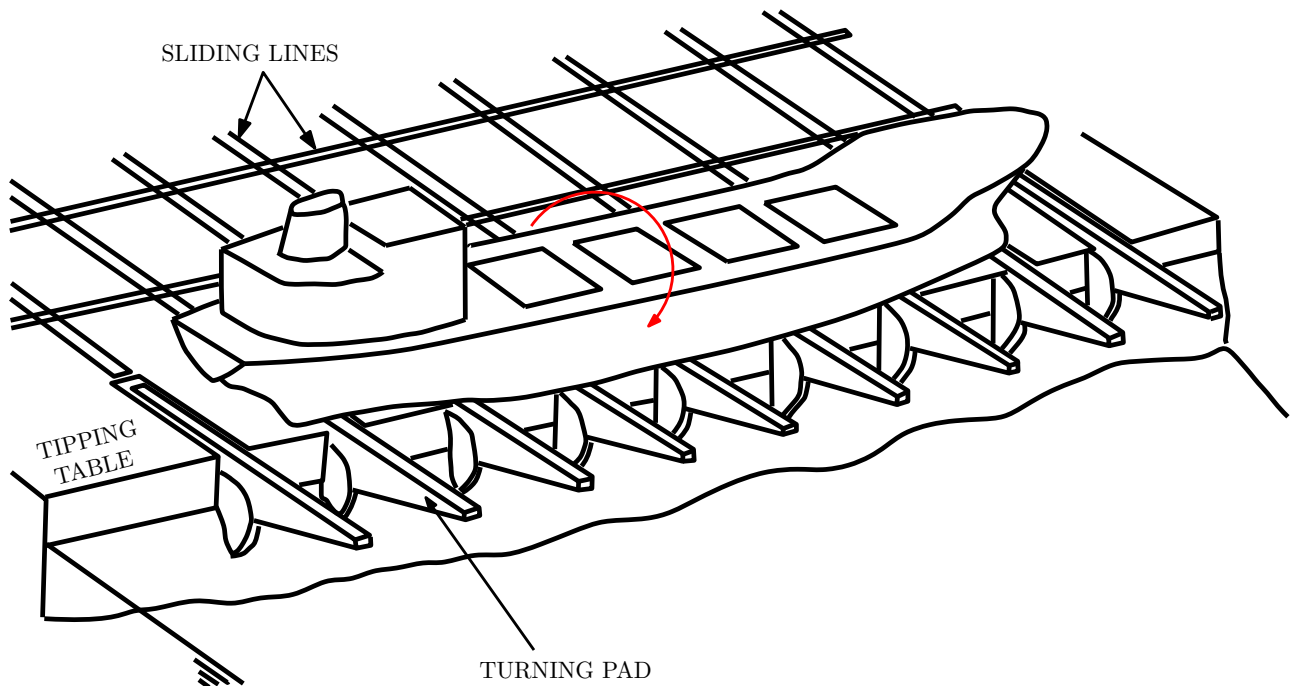


Figure 1.1: Launching system [1].

The whole process may be split into six phases where phase 3 consists of two sub-phases, i.e. 23 and 34 [2]:

- Phase 0: Pushing of structure,
- Phase 1: Structure and pad inclination in air,
- Phase 2: Structure and pad inclination in water,
- Phase 23: Structure and pad inclination and sliding of the structure along the pad,
- Phase 34: Structure and pad inclination and sliding of the structure along the pad,
- Phase 4: Structure erection in a cradle and sliding along motionless pad,
- Phase 5: Structure erection and pad inclination,
- Phase 6: Structure set afloat and return of a pad.

In this way, it is possible to launch small ships longitudinally by one or two turning pads, and large ships sideways by two or more pads. This depends on the radius of circular slipways. In the case of side launching of ships on an even keel, yawing is negligible.

1.3.1. Launching Phases

In this section, launching phases are briefly described. For further information about deriving equations of motion, initial conditions and equations of motion solution procedure, the reader is referred to [1, 2, 4, 5, 6].

Phase 0

Phase 0 or pushing of the structure is the first phase in side launching. A packed ship is pushed on the braked turning pads by hydraulic devices with a horizontal force to overcome the sliding friction force.

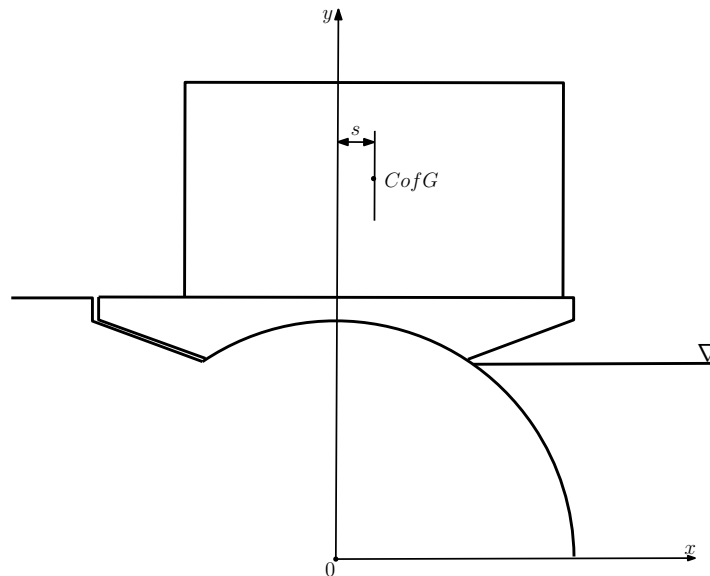


Figure 1.2: Launching system in phase 0 [2].

Phase 1

The second phase is called phase 1 or structure and pad inclination in the air. When the moment of the ship weight overcomes the friction moment of the arches about their centre, the tension forces remove the triggers, and the ship starts to rotate together with the turning pads in the air.

The relevant forces, as shown in Fig. 1.3 are ship weight D , normal reaction N and friction force T . The friction force T depends on the normal reaction N and the static

friction coefficient μ_r^0 on an arch. To analyse the ship motion, it is necessary to consider the dynamic equilibrium of the system.

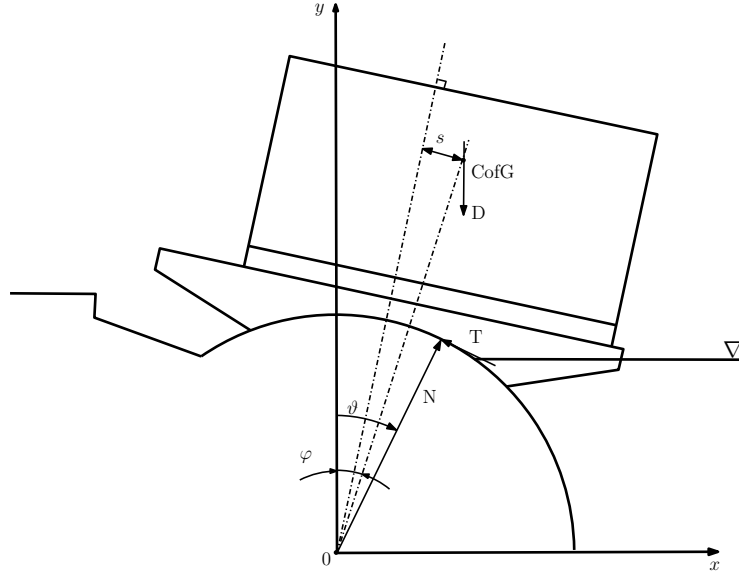


Figure 1.3: Launching system in phase 1 [2].

The basic displacements are (Figure 1.3):

- φ - pad and ship inclination angle,
- θ - ship inclination angle if $\psi \neq \varphi$,
- s - ship slide along the pad.

Ship motion is described by a system of nonlinear differential equations of the second order:

$$\ddot{\varphi} - p(\varphi, \vartheta) - q(\varphi, \vartheta) = 0, \quad (1.1)$$

and

$$\vartheta = \tan^{-1} \frac{\mu_r \tilde{A}(\varphi) \dot{\varphi}^2 - \tilde{B}(\varphi) \ddot{\varphi}}{1 + \tilde{B}(\varphi) \dot{\varphi}^2 - \tilde{A}(\varphi) \ddot{\varphi}}, \quad (1.2)$$

where \tilde{A} , \tilde{B} , p and q are coefficients depending on geometrical and physical parameters of structure and launching system, ϑ is the angle of the reaction of the circular slipway, and μ_r the dynamic value of the friction coefficient at sliding arch.

Eq. (1.1) has to be solved taking the initial conditions $\varphi = \varphi_0 = 0$ and $\dot{\varphi} = \dot{\varphi}_0$ into account, within the domain $0 \leq \varphi \leq \varphi_{1,2}$ where $\varphi_{1,2}$ is the value of the rotation angle at which the ship touches the water.

Phase 2

In phase 2 ship continues to rotate and enters into the water. Therefore, significant forces in addition to those specified in phase 1 are acting on a ship (bouyancy and hydrodynamic forces).

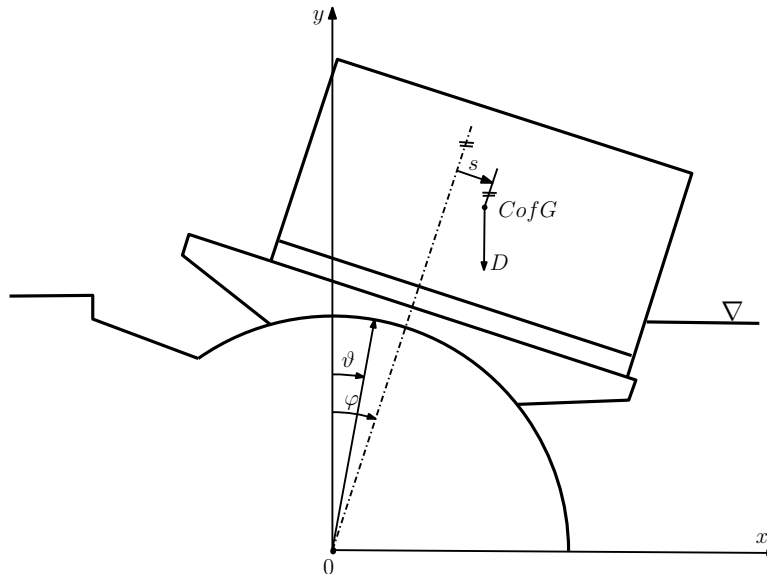


Figure 1.4: Transition from phase 1 to phase 2 [2].

Similar to the phase 1, ship motion is described by a system of nonlinear differential equations of the second order:

$$\ddot{\varphi} - p(\varphi, \vartheta) - q(\varphi, \vartheta) = 0, \quad (1.3)$$

and

$$\vartheta = \tan^{-1} \frac{\mu_r \tilde{A}(\varphi) \dot{\varphi}^2 - \tilde{B}(\varphi) \ddot{\varphi}}{1 + \tilde{C}(\varphi) \dot{\varphi}^2 - \tilde{D}(\varphi) \ddot{\varphi}}, \quad (1.4)$$

where \tilde{C} , \tilde{D} , p and q are coefficients depending on geometrical and physical parameters of structure and launching system, ϑ is the angle of the reaction of the circular slipway,

and μ_r the dynamic value of the friction coefficient at sliding arch. The initial conditions for phase 2 are $\varphi = \varphi_{1,2}$ and $\dot{\varphi} = \dot{\varphi}_{1,2}$.

Phase 3

In phase 3 the ship rotates with the pads and slips along them. First, the system inclines and then erects, so that phase 3 consists of two sub-phases, sub-phase 23 and 34 respectively.

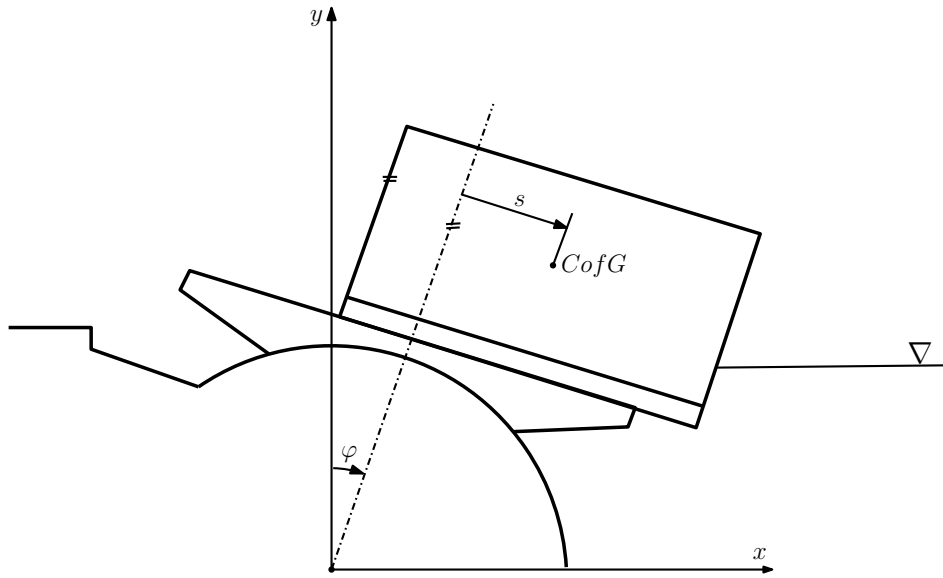


Figure 1.5: Launching system in phase 3 [2].

The ship motion is represented by a system of nonlinear differential equations of the second order:

$$p_1 \ddot{s} + p_2 \dot{s}^2 + p_3 \dot{s} \dot{\varphi} + p_4 \ddot{\varphi} + p_5 \dot{\varphi}^2 - p_6 = 0, \quad (1.5)$$

and

$$q_1 \ddot{s} + q_2 \dot{s}^2 + q_3 \dot{s} \dot{\varphi} + q_4 \ddot{\varphi} + q_5 \dot{\varphi}^2 - q_6 = 0, \quad (1.6)$$

where coefficients p_i and q_i are functions of s and φ

$$\begin{aligned} p_i &= p_i(s, \varphi), \\ q_i &= q_i(s, \varphi), \quad i = 0, 1, \dots, 6. \end{aligned} \quad (1.7)$$

Phase 4

In phase 4 the turning pads are immobile while the ship slides along them and erects in the cradles.

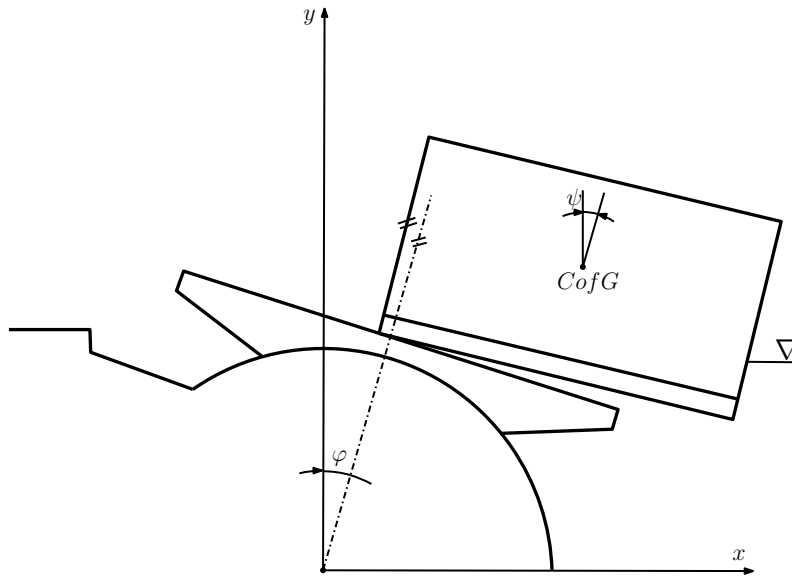


Figure 1.6: Launching system in phase 4 [2].

Equations of motion are:

$$p_1 \ddot{s} + p_2 \dot{s}^2 + p_3 \dot{s} \dot{\psi} + p_4 \ddot{\psi} + p_5 \dot{\psi}^2 - p_6 = 0, \quad (1.8)$$

and

$$q_1 \ddot{s} + q_2 \dot{s}^2 + q_3 \dot{s} \dot{\psi} + q_4 \ddot{\psi} + q_5 \dot{\psi}^2 - q_6 = 0, \quad (1.9)$$

where coefficients p_i and q_i are functions of s and φ (Eq.1.7).

Phase 5

In phase 5 cradles make a fixed contact between the ship and the pads by friction. The ship and the pads rotate in the opposite directions. The pad angle φ is increased and the ship angle ψ is decreased.

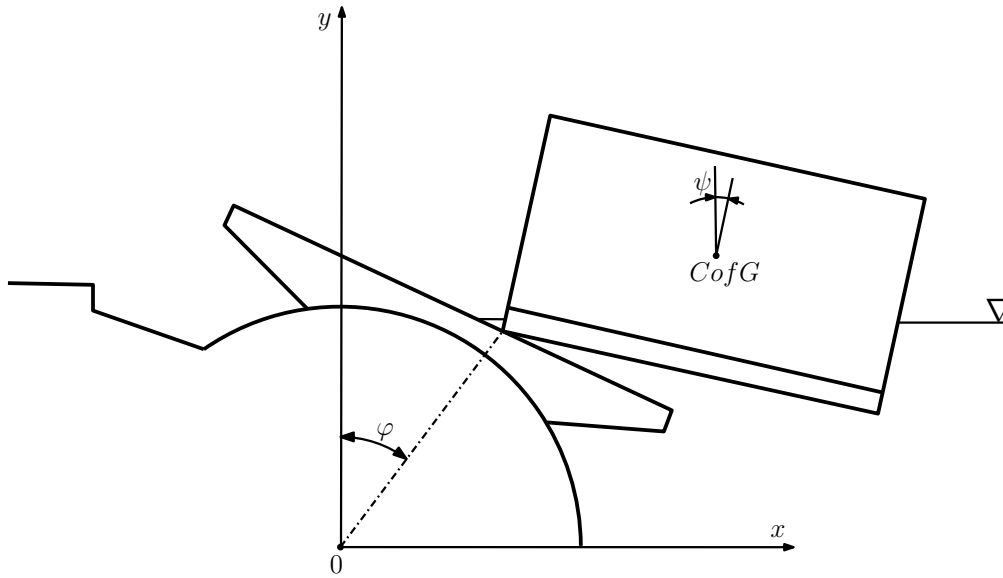


Figure 1.7: Launching system in phase 5 [2].

Equations of motion are:

$$p_1\ddot{\varphi} + p_2\dot{\varphi}^2 + p_3\dot{\varphi}\dot{\psi} + p_4\ddot{\psi} + p_5\dot{\psi}^2 - p_6 = 0, \quad (1.10)$$

and

$$q_1\ddot{\varphi} + q_2\dot{\varphi}^2 + q_3\dot{\varphi}\dot{\psi} + q_4\ddot{\psi} + q_5\dot{\psi}^2 - q_6 = 0, \quad (1.11)$$

where coefficients p_i and q_i are functions of s and φ (see Eq. 1.7).

Phase 6

Phase 6 is the final phase. As shown in Figure 1.8 the ship freely floats performing complex motion consisting of sway, heave and roll.

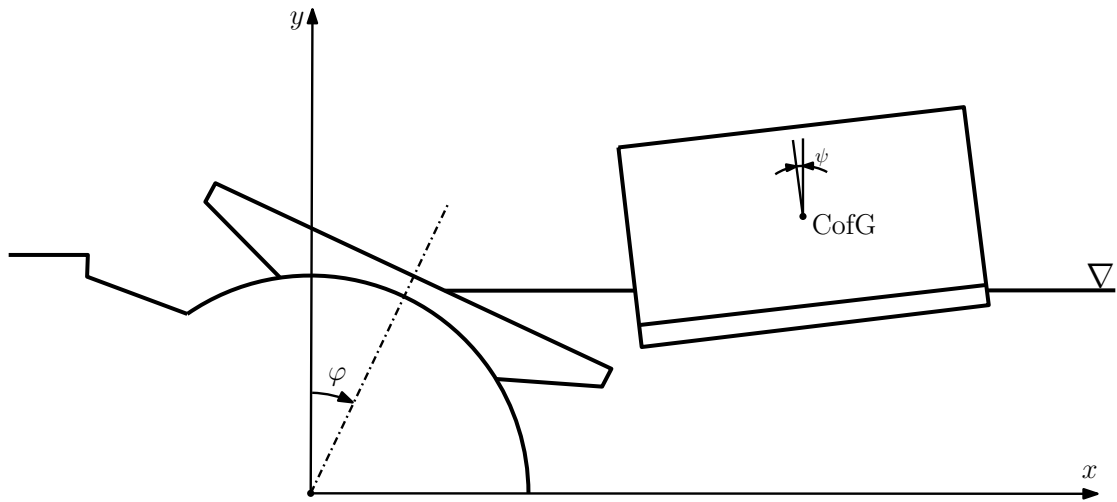


Figure 1.8: Launching system in phase 6 [2].

The coupling of motion components may be neglected and each problem may be solved as a single degree of freedom task. The equilibrium of forces leads to the differential equations (Eq. 1.12, 1.13 and 1.14) for motion components.

Sway:

$$\ddot{x} + \tilde{a}\dot{x}^2 = 0. \quad (1.12)$$

Heave:

$$\ddot{y} + \tilde{a}|\dot{y}|\dot{y} + \tilde{b}y = 0. \quad (1.13)$$

Roll:

$$\ddot{\psi} + \tilde{a}|\dot{\psi}|\dot{\psi} + \tilde{b}\psi = 0. \quad (1.14)$$

Coefficients \tilde{a} and \tilde{b} depend on the relevant drag and inertia parameters, and restoring and inertia parameters respectively [1].

1.4. Thesis Outline

Chapter 2. introduces important aspects of the overset mesh and describes overset mesh as implemented in `foam-extend` software. The `foam-extend` project is a fork of the OpenFOAM open source library for Computational Continuum Mechanics (CCM). Details of the development model of OpenFOAM and its relationship to continuum

mechanical modeling and simulations of CCM problems are provided by Weller *et. al.* [7] and Jasak *et. al.* [8],[17] Overlap fringe assembly strategies are briefly described, and the newly developed algorithm is described in detail.

Chapter 3. presents the validation of the developed algorithm as a pre-processing tool. Three test cases, with varied settings are examined: cylinder in a channel, NACA 4412 and ONR Tumblehome ship test case.

Chapter 4. presents validation of the developed algorithm, as well as the *Naval Hydro Pack* [9], with fluid flow present. Side launching of a ship simulation is performed.

Chapter 5. summarises the Thesis and gives a general conclusion.

2 | Overset Mesh Technique

In this chapter, the overset mesh technique as implemented in `foam-extend` software is described as well as newly developed automatic fringe assembly algorithm.

2.1. Introduction

Setting up an overset simulation may be divided into four main steps:

1. Generation of component meshes,
2. Hole cutting procedure,
3. Overset fringe layer assembly procedure and,
4. Overset interpolation.

Overset mesh generation is conceptually split into off-body or background meshes and near-body meshes (also called overset meshes) which resolve geometry and boundary layers [10]. The hole cutting procedure is employed to identify the cells that are outside of the computational domain. Deactivated cells are called hole cells. The hole cutting procedure is briefly described in Section 2.2. The overset fringe layer assembly procedure is employed to identify overset mesh cell types. The overset mesh cell types are introduced and described in Section 2.2. Donor/acceptor pair consists of acceptor and appropriate master donor. The donor/acceptor pairs are described in Section 2.3. Overset mesh system is composed of multiple connected domains, which are called overset regions and are introduced in Section 2.4. Fringe layer strategies are introduced in Section 2.5., while local suitability criterion is introduced in Section 2.6. Existing

fringe assembly algorithm is described in Section 2.7., while newly developed algorithm is described in Section 2.8. Finally, an overview of overset interpolation schemes and a description of the `inverseDistance` scheme, used in Chapter 4., are given in Section 2.9.

2.2. Overset Mesh Cell Types

According to their role in the solution process of the governing equations there are three base overset mesh cell types (or "overset types"):

- Acceptor cell,
- Live cell, and
- Hole cell.

Base and special overset types are presented in the form of a diagram in Figure 2.1. Arrow points from derived to a base type. The diagram should be read from bottom to top. For example, both master and extended donor are donors, while the donor is a live cell and the live cell is an overset mesh cell.

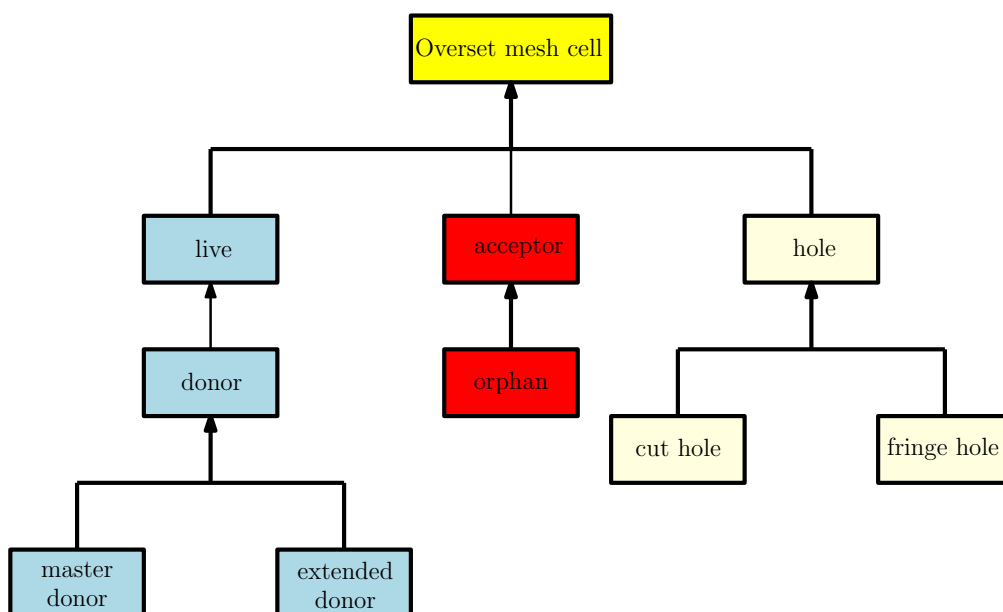


Figure 2.1: Overset mesh cell types.

Live cells are used to discretise the governing equations. Field values in those cells are obtained by solving linear system of equations.

Donor cell is a special type of live cell. It is used for overset interpolation, i.e. donor is the interpolation source for providing the value applied to the appropriate acceptor cell.

Field values at acceptor cells are obtained exclusively by interpolation from donor cells. That indicates that the field value at acceptor cell is not affected by the values of its neighbouring cells, but field values at acceptor's neighbouring cells are affected by the field value in acceptor cell. After acceptor cells are identified, donor search procedure is employed to find appropriate master donor and extended donors for all acceptors.

The extended donors are neighbours of the master donor, and together with acceptor and master donor, they form an interpolation stencil. Master donor is a cell closest to the acceptor cell. The interpolation stencil depends on the chosen overset interpolation scheme. An example of interpolation stencil is shown in Figure 2.2 where acceptor cell is denoted with A, the master donor is denoted with MD, and the extended donors are denoted with ED.

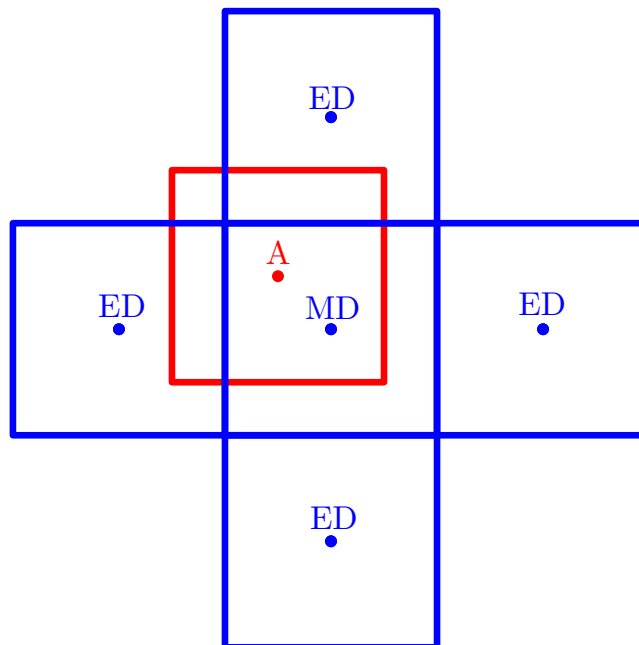


Figure 2.2: Interpolation stencil.

An orphan cell is an acceptor cell for which no valid donor is available. The orphan

cell is also called invalid donor/acceptor pair. The donor/acceptor pairs are introduced in Section 2.3. The presence of orphan cells generally indicates that there is insufficient overlap between meshes or that the mesh resolutions in overlapping region do not match well.

If the overset body meshes overlap each other, it is necessary to deactivate all regions on these meshes which are outside the computational domain, i.e. covered by other bodies or lying outside the background mesh [12]. This process is called hole cutting, and deactivated cells are called cut holes.

In Figure 2.3 four-levels overset mesh system is displayed. The mesh system contains A, B, C meshes and background mesh. Patches that determine hole cells are called hole patches. The hole patches in this system are inner patches of meshes A, B and C (walls), the outer patch of mesh A and outer patches of background mesh. The cut holes are coloured blue.

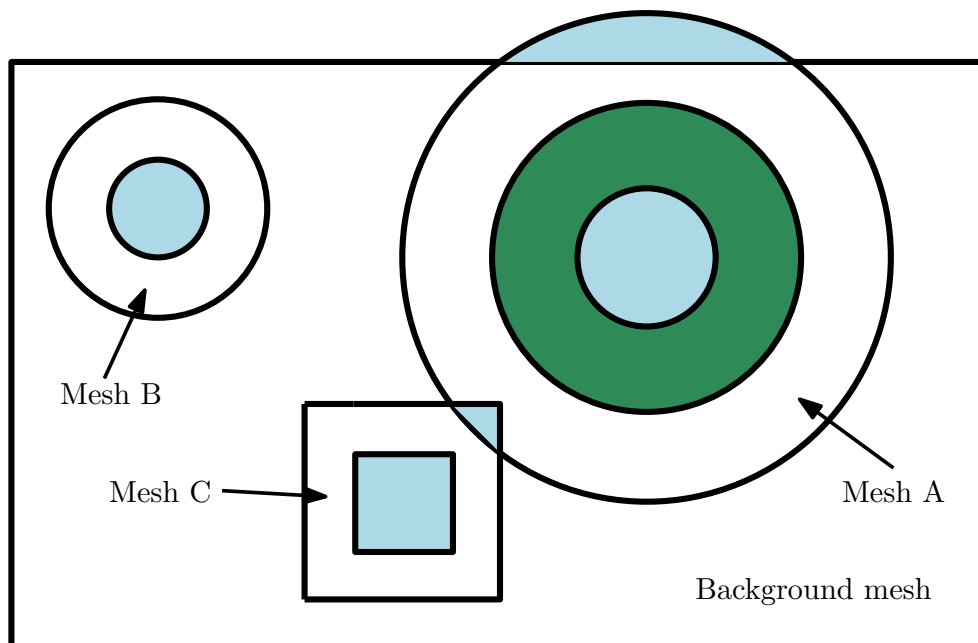


Figure 2.3: Four-levels overset mesh system.

In addition, since the parts of each mesh may be covered by another overset mesh, the holes may be created in these meshes as well [12].

During the iterative procedure, acceptors that together with donors form unsuitable donor/acceptor pairs are converted to holes in the next iteration. Hole cells which are

the result of fringe assembly procedure are called fringe holes. The fringe assembly procedures are introduced in Section 2.5. In Figure 2.3 fringe holes are coloured green.

Also, it is possible that holes or other acceptors completely surround a certain acceptor cell, so this cell needs to become a hole as well. For that purpose, filtering procedure is employed. The filtering procedure is the final step of automatic fringe assembly procedure introduced in Section 2.5.

Field value at hole cell does not depend on the field values at the neighbouring cells, nor does the hole cell field value affects the field values at the neighbouring cells. Field values at hole cells are user-prescribed, i.e. are defined in boundary conditions files using keywords `setHoleCellValue` and `holeCellValue`.

2.3. Donor/Acceptor Pairs

Inter-mesh communication is established through the fringe layer which is composed of acceptors and donors. Master donor (or shorter donor) and acceptor together form a donor/acceptor pair. The appropriate donor is identified using octree search based on eligible donors only (donor search procedure). After the hole cutting procedure, the eligible donors are cells that are not identified as cut holes nor acceptors. In the next phase, eligible donors are cells that are neither previously considered as acceptors or holes.

Each acceptor has a unique donor, while one donor can be paired with multiple different acceptors. Pairing one donor with a number of acceptors may lead to an unphysical overall solution, which occurs when meshes do not match in resolution. Therefore, in the overlapping zone, cells should be of comparable size in both meshes. Also, the coarser of the two coupled meshes determines the interpolation error level.

In `foam-extend` software, the `donorAcceptor` type represents individual donor/acceptor pair. It holds donor and acceptor data:

- Cell ID,
- Processor number,
- Cell centre position vector,
- Boolean variable which stores information whether the cell centre of the master donor falls within the acceptor cell.

The boolean variable is an important criterion for choosing the most suitable overlap assembly in `adaptiveOverlap` fringe assembly algorithm which is introduced in Section 2.8. It provides information whether the cell centre of the master donor falls within the acceptor cell. The most important `donorAcceptor` class member functions are used to set donor and to set extended donors.

The `overlap` fringe algorithm permits a presence of orphan cells. Donor/acceptor pair is being set exclusively if the cell centre of the master donor falls within the acceptor cell, i.e. acceptor is not an orphan. The `adaptiveOverlap` algorithm does not permit a presence of orphan cells, i.e. master donor is an eligible cell whose cell center is the closest to acceptor's cell centre.

2.4. Overset Regions

As stated in the Section 1.1., overset mesh system is composed of multiple connected domains, i.e. computational meshes. In `foam-extend` software, those domains are called overset regions and are presented by `oversetRegion` type which `oversetRegion`:

- Collects its cells by cell zone,
- Contains a list of donor regions,
- Contains a list of patch names used to define a hole (walls),
- Contains the fringe selection algorithm,
- Provides data on cells within a region,
- Provides data on local acceptor, donor and hole cells,
- Provides data on eligible local donors.

The `oversetRegion` also controls the iterative process of fringe assembly with the help of fringe assembly algorithm. The flowchart of the algorithm is:

1. Someone (usually `oversetMesh`'s class member function) requests the region data for donors and acceptors, which triggers the lazy evaluation in `calcDonorAcceptorCells()` member function. During the donor/acceptor assembly, all regions are interdependent, because eligible donors and holes could change during the iterative process. Hence, asking a certain region for its donors/acceptors triggers the fringe assembly in all regions.

2. The iterative process starts by going through each region, updating current donor/acceptors (`updateDonorAcceptors()` member function) which does the following:
 - (a) Asks fringe handler a set of new acceptors,
 - (b) Finds donors for these acceptors,
 - (c) Transfers donor/acceptor pairs to the fringe assembly algorithm, which decides whether a suitable overlap has been found.
3. If the suitable overlap has been found for all regions, then the donor/acceptor pairs are being finalised by calling `finaliseDonorAcceptors()` member function, which allocates requested fields (donor and acceptor cells fields).

2.5. Overlap Fringe Assembly Strategies

As stated in Section 2.4., iterative procedure is being controlled with the help of fringe assembly algorithm. In this section available fringe assembly strategies in `foam-extend` software are presented. There are different fringe assembly variants:

- `manual` - user provides a cell set for acceptor cells,
- `faceCells` - acceptors are face cells of a given patch,
- `overlap` - the most suitable overlap is found automatically using an iterative neighbourhood search algorithm,
- `adaptiveOverlap` - improved version of the overlap procedure presented in this work,
- `composite` - consists of multiple fringes on a single region.

The algorithms are represented by related classes which are derived from `oversetFringe` abstract class. Class diagram is presented in Figure 2.4.

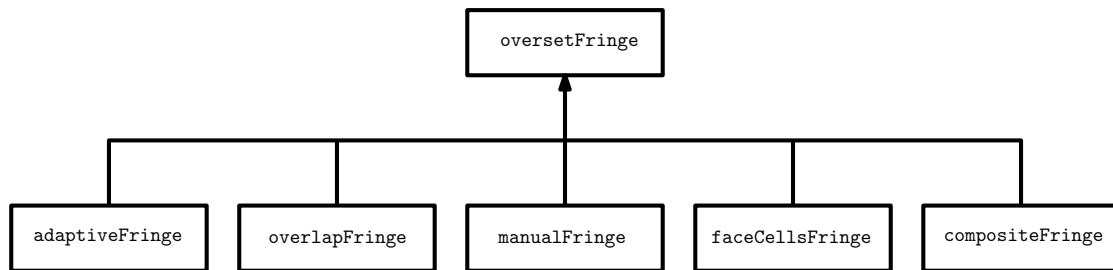


Figure 2.4: Fringe algorithms class diagram.

Defining a cell set for acceptor cells manually can represent an overhead from engineer’s perspective and would be a difficult task if overset mesh system contains a larger number of complex meshes.

The most commonly used strategy is automatic `overlap` algorithm because of its ease of use, as well as `faceCells` variant. If the `faceCells` is employed, acceptors are face cells of a given patch. Therefore, the `faceCells` variant cannot be used if the given patch of body mesh lies outside the background mesh (see Figure 2.3). In this Thesis, `adaptiveOverlap` is developed as an improvement to the `overlapFringe`, which determines the fringe layer more robustly and reliably.

The fringe assembly algorithm:

- Is owned by a particular `oversetRegion`,
- Defines the hole cells for its region,
- Defines acceptor cells for its region,
- Controls the iterative process for finding suitable overlap,
- Defines donor/acceptor list used by `oversetRegion` to create combined data.

Automatic overlap fringe assembly algorithms are described in detail in Sections 2.7. and 2.8. The first step in automatic fringe procedures is called addressing calculation procedure and it is described in Subsection 2.5.1.

2.5.1. The calcAddressing Procedure

The first step in fringe assembly procedure is to get an initial guess for holes and acceptors. For that purpose `calcAddressing()` member function is employed. Neighbourhood search starts from hole neighbouring cells and optionally from specified patches

(initial patches). This step is common to `overlapFringe` and `adaptiveOverlapFringe` algorithms.

After the hole cutting procedure is completed, `calcAddressing()` identifies neighbouring cells of cut holes, which are eligible, as acceptors. If an initial patch is specified, it identifies eligible face cells as acceptors. At this moment eligible cells are the cells that are not cut holes. The flowchart of the algorithm is:

Algorithm 1 The `calcAddressing` procedure

- 1: If they are present, get cut holes list from current overset region.
 - 2: Initialise `eligibleAcceptors` boolean list for eligible acceptors with true.
 - 3: Loop through cut holes list and mark all cut holes as ineligible.
 - 4: Construct dynamic list `candidateAcceptors` for storage of acceptors.
 - 5: Loop through cut holes neighbouring cells and find acceptor candidates, i.e. eligible cells. Append acceptor candidates to `candidateAcceptors` dynamic list.
 - 6: Loop through optionally user-specified patches and mark face cells as candidate acceptors if are eligible. Append acceptor candidates to `candidateAcceptors` dynamic list.
 - 7: Allocate and populate data member acceptors and holes lists.
-

After the `calcAddressing` procedure is completed, the first guess for acceptors that is used as an initial condition for the iterative overlap assembly process is obtained. For each acceptor cell, master donor cell on the mesh that overlaps it has to be identified, as well as extended donors. For that purpose, donor search procedure is employed. After donors are found, it is necessary to check if suitability criteria are satisfied. If they are satisfied, the iterative procedure for current region is finished. Otherwise, a new set of acceptors should be found. Donor/acceptor suitability criteria are introduced in [Section 2.6](#).

2.6. Donor Suitability

Cells that form donor/acceptor pair have to be similar by some criterion which is called local suitability criterion.

Automatic fringe assembly algorithm controls the iterative process in a manner that it receives donor/acceptor pairs, checks if they are suitable (local suitability criterion)

and calculates average suitability of all donor/acceptor pairs. If the global criterion is reached, the iterative procedure is finished, and donor/acceptor pairs are filtered. In this section, the local suitability criterion is discussed.

Finding suitable master donor/acceptor pairs in automatic fringe assembly algorithms is based on run-time selectable Donor Suitability Functions (DSFs). In `foam-extend` available DSFs are:

- `noSuitability`: master donor cell is any cell as long as the cell centre of the master donor falls within the acceptor cell,
- `patchDistance`: pushes the overlap at a certain distance from the patches,
- `cellVolumes`: cells with similar volumes are preferred,
- `faceArea`: cells with similar minimum face area are preferred,
- `cellBoundingBoxDiagonal`: cells with similar bounding box diagonals are preferred.

DSFs cannot be used if the donor is not set, i.e. in the case of the orphan presence. The most common used DSF is `cellVolumes`. The DSFs provide information whether the donor is suitable or not. The `adaptiveOverlap` algorithm uses modified DSFs. Modified DSFs return suitability fraction value which gives more realistic insight in cells similarity. Suitability fraction (or shorter suitability) is a positive value ($S \in \langle 0, 1 \rangle$). A higher value of suitability means better suitability, i.e. that cells are more similar by some of the criteria.

Suitability is calculated as a relative difference of acceptor and donor cells properties. If $a_{dsf} > d_{dsf}$, then

$$S = 1 - \frac{a_{dsf} - d_{dsf}}{a_{dsf} + SMALL}, \quad (2.1)$$

where S stands for suitability, a_{dsf} stands for acceptor's property, d_{dsf} stands for donor's property and $SMALL$ stands for a small number 10^{-16} in order to avoid division by zero.

Otherwise

$$S = 1 - \frac{d_{dsf} - a_{dsf}}{d_{dsf} + SMALL}. \quad (2.2)$$

If `cellVolumes` DSF is employed, a_{dsf} is acceptor's volume and d_{dsf} is donor's volume and S is their relative difference.

2.7. The overlap Algorithm

After the initial guess for acceptors (as a result of the `calcAddressing` procedure) is obtained and donor/acceptor pairs are set, it is necessary to check if donor/acceptor pairs satisfy two criteria. The iterative `overlap` assembly is controlled with two parameters [18]:

- `threshold` for DSF (local suitability criterion): the maximum allowable difference between candidate donor's and acceptor's DSF:
 - e.g. 0.5 means that donor/acceptor cell volumes should be within 50% of the larger cell among the two.
- `suitablePairFracion`: global stopping criteria for the iterative process:
 - e.g. 0.8 means that the iterative process stops when 80% of all donor/acceptor pairs are suitable according to DSF.

The `overlap` algorithm receives the donor/acceptor list and loops through all list items, checks if they are orphans, suitable or unsuitable, counts how many pairs are suitable according to DSF and counts orphans, if they are present.

The suitable donor/acceptors pairs are appended into cumulative donor/acceptor list and algorithm holds them until the end of the iterative process. The unsuitable donor/acceptor pairs are appended into unsuitable donor/acceptor pairs list. After it has looped through all donor/acceptor pairs, it calculates percentage of suitable pairs in the list.

If there are no orphans present and the global stopping criterion is reached, the iterative procedure is finished. Filtering procedure is employed, i.e. the algorithm checks if there is any acceptor present which is surrounded by other acceptors or holes. If it is, filtering procedure converts it to a hole.

Otherwise, new acceptors should be found. Acceptor candidates are neighbouring cells of acceptors from unsuitable donor/acceptor pairs which are not holes nor previously considered as acceptors. Algorithm loops through unsuitable donor/acceptor pairs list, converts acceptors from unsuitable donor/acceptor pairs to holes and marks all acceptor candidates as new acceptors.

2.7.1. Disadvantages of the overlap Algorithm

The main disadvantage of the `overlap` algorithm is that it does not track iteration history, i.e. it just propagates ahead and does not have the capability to take fringe assembly from an earlier iteration (i.e. does not have a fall-back mechanism), if its average suitability is higher than the suitability of fringe assembly from the last iteration.

From engineer's perspective, it is a hard task to estimate `overlap` algorithm parameters, especially if overset mesh system contains a larger number of 3D meshes.

Also, it has ill-defined stopping criteria. The most common case is that user overestimates parameters, i.e. provides unrealistically high values. For example, let's consider an extreme case where provided value for `threshold` and for `suitablePairFraction` is 1, while employed DSF is `cellVolumes`. This means that cells that form every donor/acceptor pair have to be equal in volume. It is clear that this is almost impossible to achieve, except in the case of equal meshes, so algorithm will propagate forward until unsuitable acceptor's neighbours are ineligible cells leading to failure of the algorithm.

Overestimation of parameters may lead to the failure and underestimation of parameters may lead to a low-quality fringe assembly which may lead to high interpolation errors or unphysical overall solution.

To achieve a high quality overset fringe assembly, a user has to vary `overlap` parameters, which represents an often unnecessary overhead from its perspective. For that purpose the `adaptiveOverlap` algorithm, which calculates average suitability more sensitively, tracks iteration history, has a better defined stopping criterion and fall-back

mechanism is developed. The algorithm is introduced in Section 2.8.

2.8. The adaptiveOverlap algorithm

For the purpose of this Thesis, the automatic `adaptiveOverlap` algorithm for overset meshes is developed as an improvement to the `overlap` algorithm.

The `adaptiveOverlap` is controlled by four parameters and all of them have default values. The parameters together with their data types and default values are presented in Table 2.1. Among those parameters, user should choose the DSF function. The scalar data type is equivalent to standard C++'s `double` and `label` to `int`.

Table 2.1: Algorithm parameters.

Parameter:	Data type:	Default value:
<code>minLocalSuit</code>	scalar	0
<code>specifiedIterationsNumber</code>	label	4
<code>additionalIterations</code>	Switch	true
<code>orphanSuitability</code>	scalar	-100

Here, the orphan has a different meaning. Orphan is an acceptor cell for which donor is found, but its cell centre does not fall within acceptor's bounding box. Presence of orphans is undesirable because if donor and acceptor, which form a single donor/acceptor pair, are far apart, unphysical overall solution can be obtained.

Orphan suitability is not calculated in a standard manner using DSFs - it is user prescribed. By prescribing negative suitability to an orphan, average suitability is being degraded. That reduces a possibility that fringe assembly with an orphan present is being chosen for the final overset assembly. By default, `orphanSuitability` is -100% which means that the chosen donor is not suitable at all.

Following the `overlap` algorithm pattern, after an initial guess from the *calcAddressing* procedure is obtained, the algorithm loops through the received donor/acceptor list, checks if an orphan is present and if it is, adds its suitability to unsuitable pairs cumulative suitability, increment orphan counter and appends it to unsuitable donor/acceptor pairs list. Otherwise, it calculates donor/acceptor suitability using DSF.

Pairs for which suitability is lower than minimum local suitability are categorised as unsuitable while pairs for which suitability is higher than minimum local suitability are categorised as suitable. After looping through the entire list, average suitability of all pairs is calculated

The `adaptiveOverlap` tracks iteration history, i.e. after a single iteration is made, it creates object of an `iterationData` class and pushes it onto the `FIFOStack` (First In First Out) container named `iterationDataHistory`. The `iterationData` class holds as data members everything that is needed to reconstruct fringe layer assembly from a specific iteration and to calculate slope using linear regression:

- List of donor/acceptor pairs,
- List of hole cells,
- Average suitability and,
- Iteration number.

After user-defined number of iterations is made (`specifiedIterationsNumber`) and if user-defined `additionalIterations` as true, algorithm approaches data set using linear regression - it calculates suitability slope coefficient, i.e. suitability gradient. Here average suitability is the predicted variable and iteration number is the predictor variable.

It is assumed that if the gradient is positive, fringe assembly with greater average suitability may be found, so one additional iteration will be made (relative iteration counter is incremented). The bottom element of `iterationDataHistory` is being popped of the stack. In that way there are always `specifiedIterationsNumber`, or one less, number of elements in the stack.

In Figure 2.5, average suitability against iteration number diagram is shown. After four iterations were made, the suitability gradient was calculated. Due to positive suitability gradient, the first element, i.e. iteration 1, was popped off the container. Then there were three objects in the container. One additional iteration was made, i.e. iteration 5. After iteration 5 was completed, due to negative suitability gradient, the iterative procedure was terminated. For final overset fringe assembly, the iteration with the highest average suitability value was chosen, i.e. iteration 4.

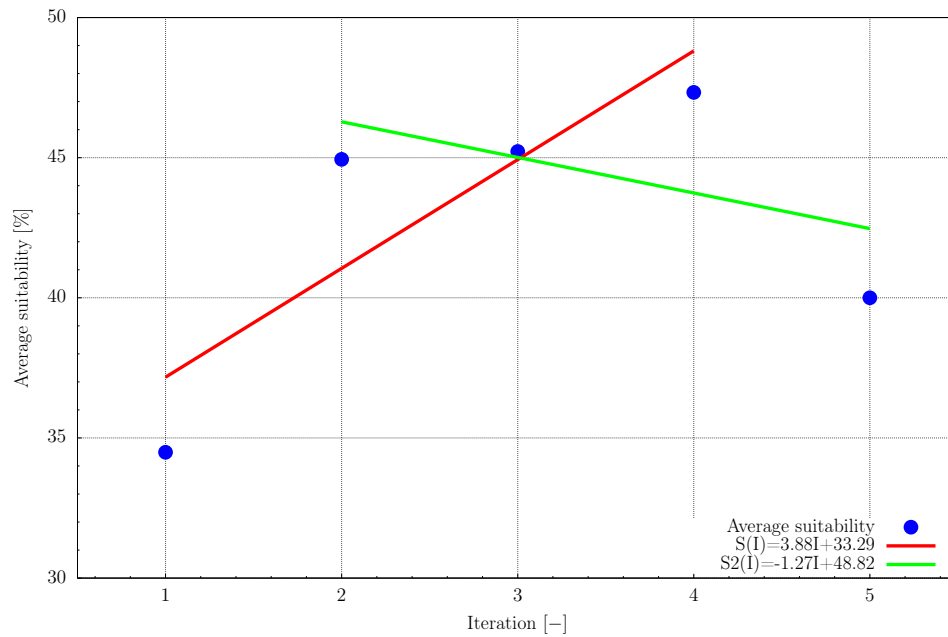


Figure 2.5: Iterative procedure diagram.

The algorithm checks if the unsuitable pairs list is empty and if a desired number of iterations is made (global criterion is satisfied). If those two criteria are not reached, it finds a new set of acceptors. The new acceptors are immediate neighbours of unsuitable acceptors which are eligible (are not holes nor acceptors from current iteration). Acceptors from unsuitable donor/acceptor pairs are converted to holes, i.e. are appended to the holes list.

Otherwise, the iterative procedure is over. To summarise, if default values for parameters are used, the iterative procedure is over if:

- User-defined criteria have been reached, i.e. there is no more unsuitable donor/acceptor pairs or,
- `specifiedIterationsNumber` number of iterations are made and the slope is negative or,
- After additional iteration is made and the slope is negative.

When the iterative process is over, the algorithm finds an iteration with the highest average suitability from `iterationDataHistory` container.

The final step is pairs filtering: it is possible that holes or other acceptors completely surround a certain acceptor cell, so this cell needs to become a hole as well. Flowchart of the algorithm is given in Figures 2.6 and 2.7.

Now overset types are identified and class's `oversetMeshAddressing` member function `calcDomainMarkup()` constructs volume scalar field for all overset regions with value equal to zero for all elements. The zero value represents live cells. Field values for other overset types are presented in Table 2.2.

Table 2.2: Overset types field values.

Overset type	Field value
acceptor	2
donor	1
live	0
hole	-1

The `calcDomainMarkup()` first marks the donor cells, then it marks the acceptor cells and finally the hole cells. Marking order is important only for the `overlap` algorithm which holds donor/acceptor pairs from all iterations. For the `adaptiveOverlap` marking order is not important because it holds only suitable donor/acceptor pairs.

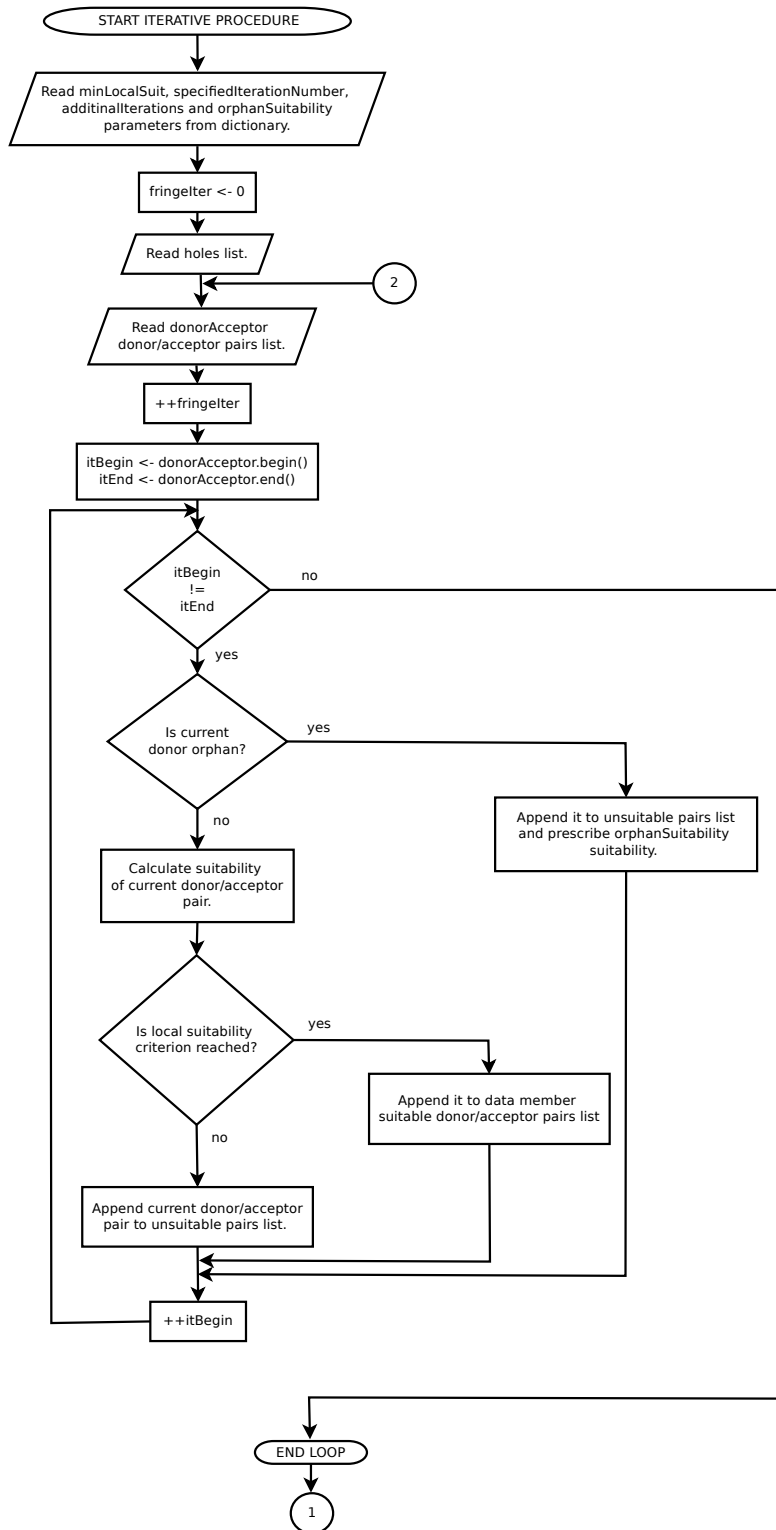


Figure 2.6: Flow chart of the adaptiveOverlap (part 1).

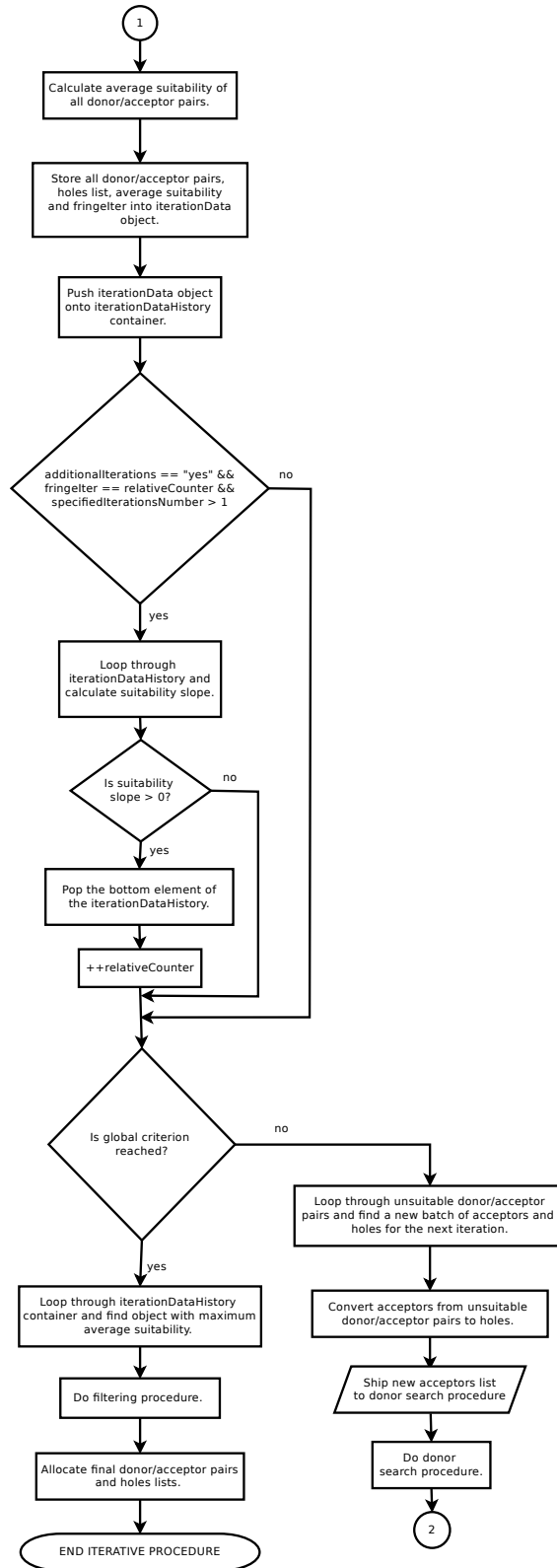


Figure 2.7: Flow chart of the adaptiveOverlap (part 2).

2.9. Overset Interpolation Schemes

The overset interpolation can be carried out implicitly or explicitly (run-time choice) [19]:

- Implicit overset: interpolation from donors to acceptors is performed in each linear system solver interpolation, yielding strong coupling between regions,
- Explicit overset: interpolation from donors to acceptors is performed only to set the values before the solution of equations, yielding weak coupling between regions and requiring additional outer iterations.

There are three most common used run-time selectable interpolation schemes:

1. `injection`: first order, bounded,
2. `averageValue`: first order, bounded,
3. `inverseDistance`: first/second order, bounded.

The `injection` interpolation scheme takes the master donor field value and injects it into acceptor's field value. The `averageValue` scheme calculates acceptor's field value as average field value of master and extended donors. The `inverseDistance` scheme is used in Chapter 2.4. The flowchart is:

Algorithm 2 Inverse distance interpolation procedure

```

for All overset regions do
2:   Get acceptors list for current region.
   Get weights field w for current region.
4:   for All donor/acceptor pairs from current region do
       Get current donor/acceptor pair.
6:       Get acceptor from current donor/acceptor pair.
       Calculate total number of donors for current acceptor (master donor + ex-
tended donors)
8:       Initialize weight field w to one for donors of current acceptor.
       Get acceptor's cell centre position vector accCC.
10:      Calculate master donor weight as  $w_i = \frac{w_i}{\|\mathbf{accCC} - \mathbf{dCC}\|^2 + SMALL}$ 
       for Every extended donor of current acceptor do
12:          Calculate weight a  $w_i = \frac{w_i}{\|\mathbf{accCC} - \mathbf{dCC}\|^2 + SMALL}$ 
       end for
14:   end for
end for

```

Here **dCC** stands for donor's cell centre position vector. To establish overset interpolation, the outer boundary of the overset body is set to overset mesh boundary condition.

2.10. Closure

This chapter describes overset mesh technique as implemented in **foam-extend** software. Furthermore, it describes the newly developed automatic algorithm for fringe assembly. The next chapter is dedicated to testing of the algorithm as a pre-processing tool.

3 | Overset Assembly Test Cases

In this chapter, the validity of the implemented algorithm is tested on three test cases: cylinder in a channel, NACA 4412 and ONR Tumblehome test case. Capability to produce both static and dynamic overset fringe assemblies is tested. Stopping criterion and capability to choose the most suitable overlap assembly is tested with extreme values for minimum local suitability criterion.

3.1. Two-Levels Mesh of Cylinder In a Channel

In this section, the algorithm is tested on the cylinder in channel overset case with standard mesh and mesh with inverse cell grading. Static and dynamic overset assemblies are obtained. The algorithm settings are shown in Table 3.2. The `minLocalSuit` parameter is varied, i.e. the algorithm is tested with extreme values for parameter (0 and 100).

The geometry of the computational domain is shown in Figure 3.1. All the cells in the *background* region are equal in size. For the setups from Subsections 3.1.1., 3.1.2. and 3.1.4., cells in the *front* region mesh increase in size as the radius of cylinder increases. Combined meshes with detail view are shown in Figure 3.2. For the setup from Subsection 3.1.3., grading is changed in a way that cells decrease in size as the radius of cylinder increases, what is an important test for the algorithm stopping criterion. Combined meshes with detail view are shown in Figure 3.3.

As it is presented in Table 3.1, the cylinder in channel test case consists of 2 overset regions, *front* and *background* regions. The *front* region consists of 30 000 cells, while

the *background* region consists of 12 800 cells. Overall number of cells is 42 800 and all the cells are hexahedral.

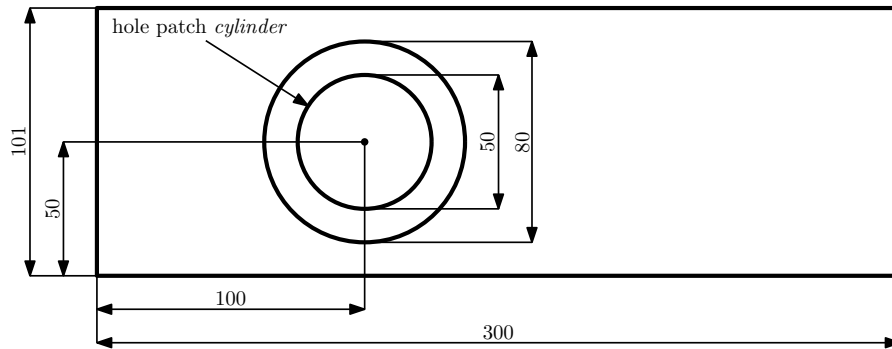


Figure 3.1: Geometry of the computational domain for the 2-D cylinder in a channel test case.

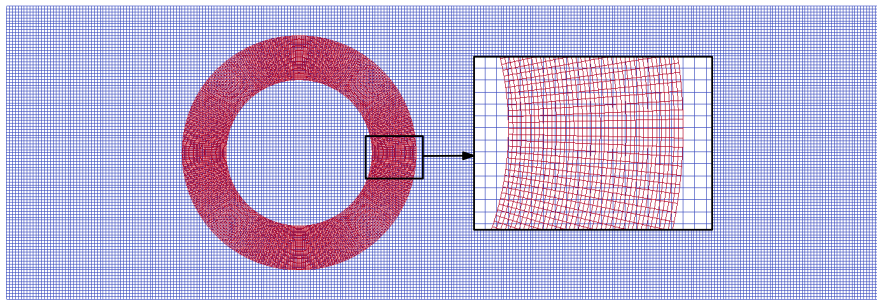


Figure 3.2: Combined meshes with detailed view (standard grading).

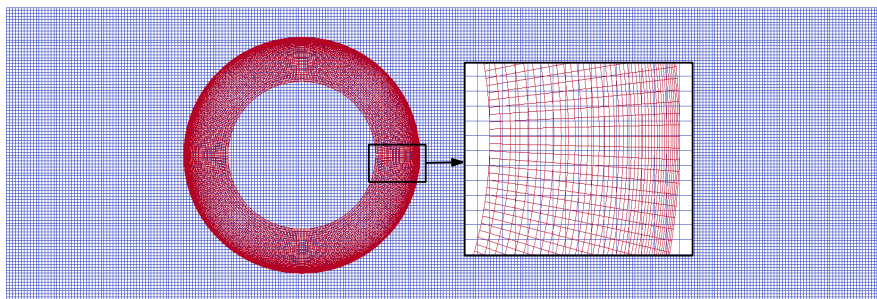


Figure 3.3: Combined meshes with detailed view (inverse grading).

Table 3.1: Overset mesh system statistics.

Number of regions:	2	
Region name:	<i>front</i>	<i>background</i>
Number of cells per region:	30 000	12 800
Overall number of cells:	42 800	

Table 3.2: Fringe assembly settings.

Region name:	<i>background</i>	<i>front</i>
Fringe type:	<code>adaptiveOverlap</code>	<code>faceCells</code>
Donor regions:	<i>front</i>	<i>background</i>
<code>initPatchNames:</code>	/	/
<code>minLocalSuit:</code>	case-dependent	/
<code>specifiedIterationsNumber:</code>	4	/
<code>additionalIterations:</code>	true	/
<code>orphanSuitability:</code>	-100	/
DSF:	<code>cellVolumes</code>	/
<code>holePatches:</code>	cylinder	
<code>patches:</code>	/	outerCylinder

3.1.1. Static Overset Assembly Test Case Setup 1 (standard grading)

The first test case is performed with standard *front* region mesh (Figure 3.2) and with the minimum value for minimum local suitability parameter (`minLocalSuit=0%`).

In Table 3.3 relevant iteration data is provided: iteration number, number of unsuitable pairs (the first number stands for number of the orphans, while the second number stands for donor/acceptor pairs whose suitability is lower than user-prescribed minimum local suitability), average donor/acceptor pairs suitability for each iteration and suitability slope which is calculated after 4 iterations were made or after additional iteration is made.

As it can be seen from Table 3.3, the iterative procedure was finished in 1 iteration because unsuitable donor/acceptor pairs were not found. Overset types field for each

region is shown in Figure 3.4. Final overset mesh assembly is shown in Figure 3.5.

Table 3.3: Iteration data.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	0/0	18.76	/

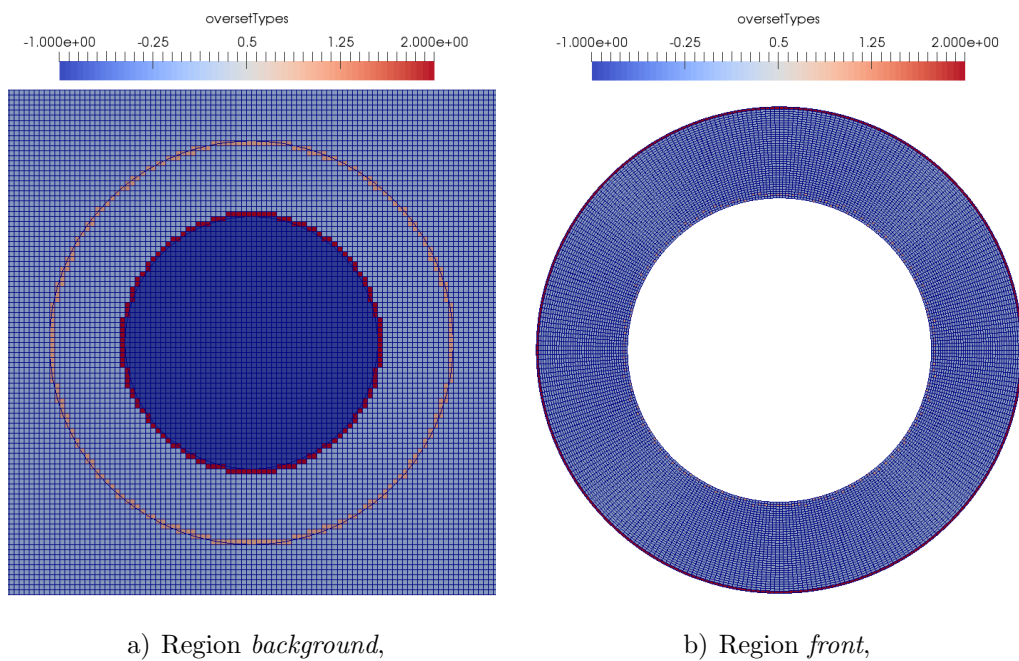


Figure 3.4: Overset types.

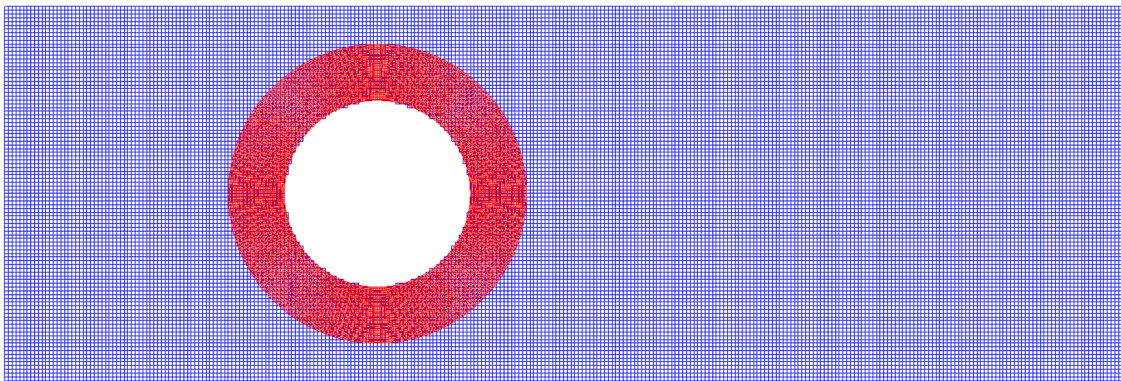


Figure 3.5: Final overset assembly.

This test case is also performed with different values for orphan suitability parameter. It was concluded that the value of `orphanSuitability = -7%` was appropriate to degrade suitability slope and to stop iterative procedure.

3.1.2. Static Overset Assembly Test Case Setup 2 (standard grading)

The second test case is performed with standard *front* region mesh (Figure 3.2) and with the maximum value for minimum local suitability criterion (`minLocalSuit=100%`).

The main goal of this case is to test the algorithm stopping criterion and the fall-back mechanism, i.e. ability to choose the most suitable overlap assembly.

As can be seen from Table 3.4 average suitability continually increases and additional iterations were made. In the iteration 15, 12 orphans were found. That occurred because in iteration 15 there were no more eligible acceptors.

Presence of unsuitable acceptors degraded average suitability and the calculated suitability slope was negative. The iterative procedure stopped due to the negative suitability slope and the overset assembly with the highest average suitability value, i.e. assembly from iteration 14, was chosen.

Final overset types fields for all regions are shown in Figures 3.6, while final overset assembly is shown in Figure 3.7.

Table 3.4: Iteration data.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	0 / 144	18.76	/
2	0 / 148	19.45	/
3	0 / 152	20.08	/
4	0 / 156	20.77	0.66
5	0 / 160	21.41	0.65
6	0 / 164	22.05	0.65
7	0 / 168	22.73	0.65
8	0 / 172	23.36	0.65
9	0 / 176	24.01	0.65
10	0 / 180	24.67	0.64
11	0 / 184	25.32	0.65
12	0 / 188	25.94	0.64
13	0 / 192	26.61	0.64
14	0 / 196	27.21	0.63
15	12 / 188	20.10	-1.69

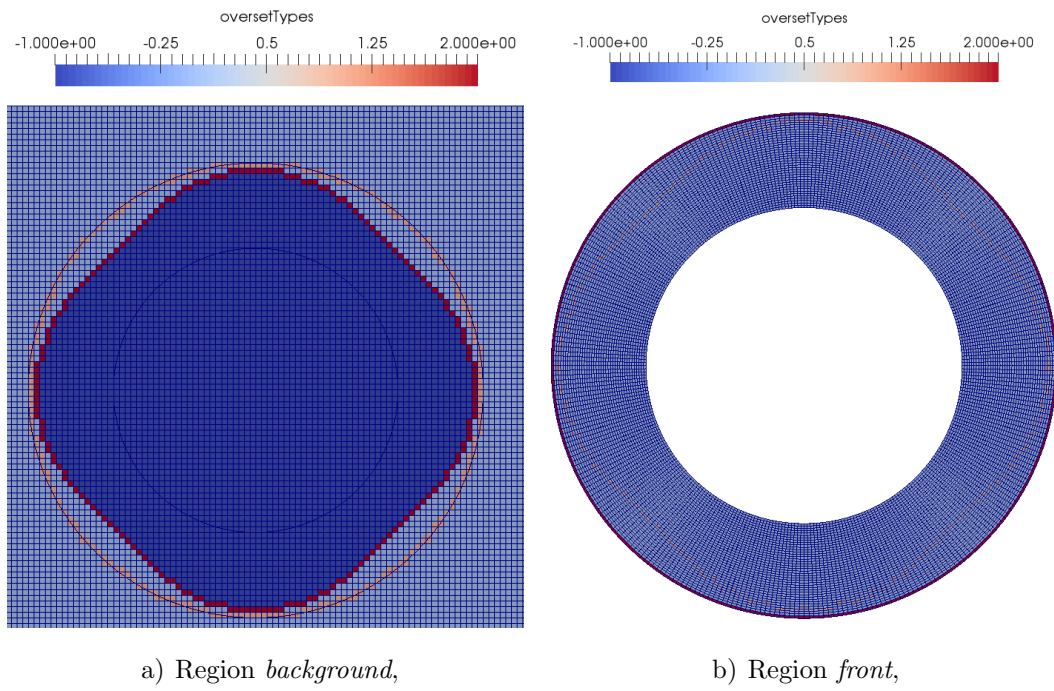


Figure 3.6: Overset types.

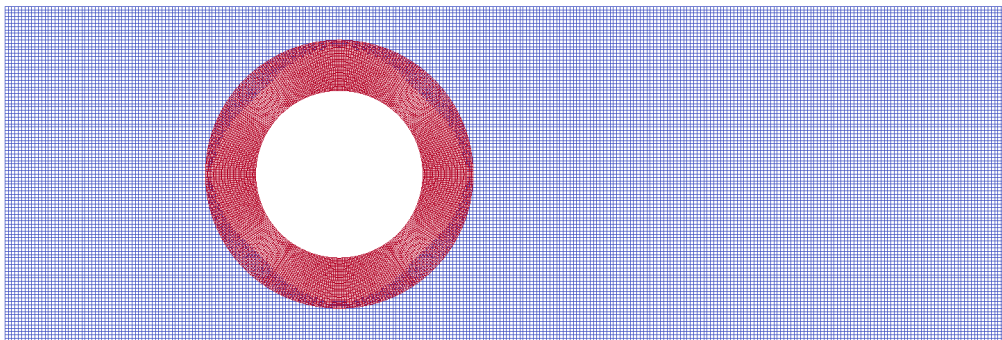


Figure 3.7: Final overset assembly detailed view.

3.1.3. Static Overset Assembly Test Case Setup 3 (inverse grading)

The third test case is performed with *front* region mesh with inverse grading (Figure 3.3) and with maximum value for minimum local suitability ($\text{minLocalSuit}=100\%$).

As it can be seen from Table 3.5, average suitability continually decreases. After 4 iterations were made, iterative procedure stopped due to the negative suitability slope. The first iteration was chosen for final overset mesh assembly.

Table 3.5: Iteration data.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	0 / 144	41.36	/
2	0 / 148	40.5	/
3	0 / 152	39.52	/
4	0 / 156	38.43	-0.97

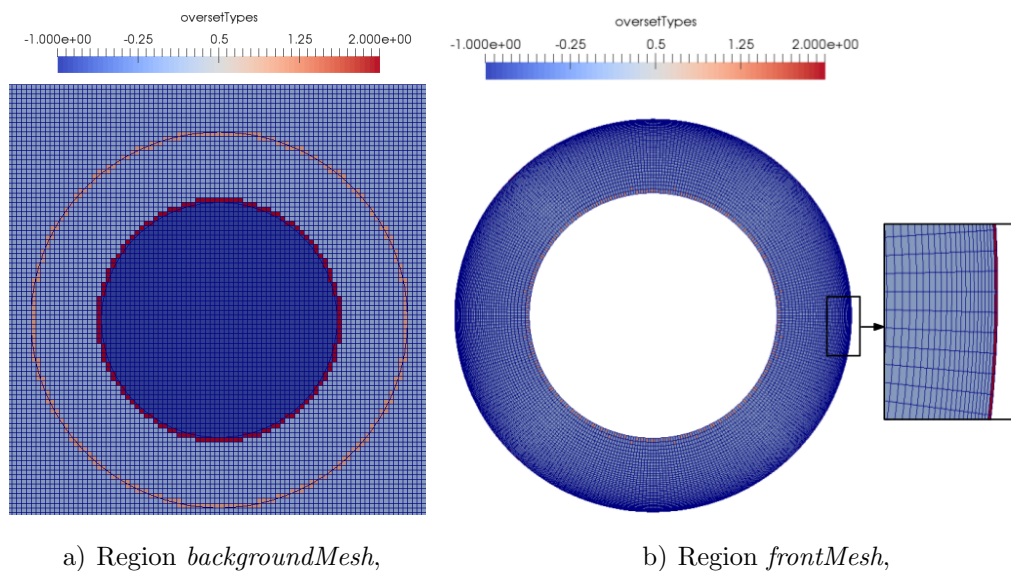


Figure 3.8: Iteration 1.

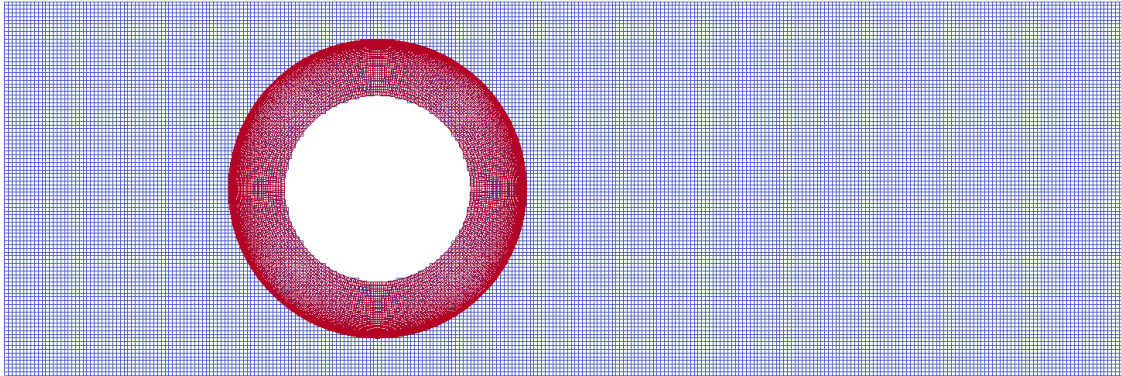
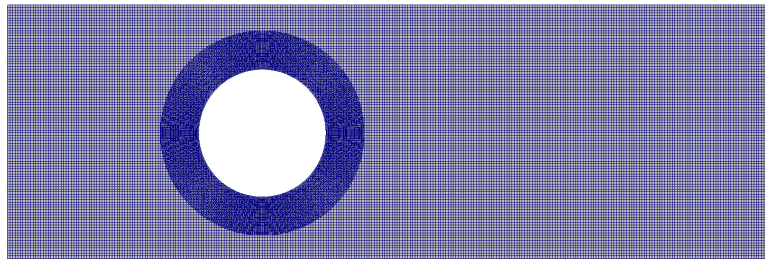
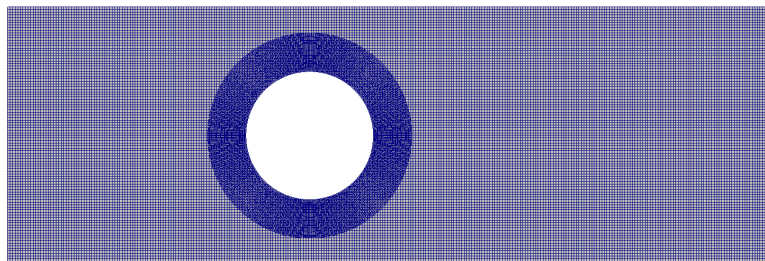
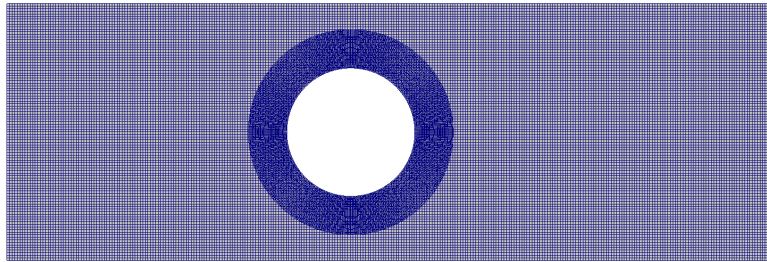
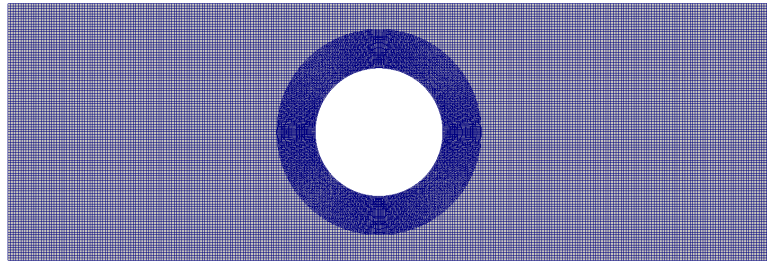


Figure 3.9: Final assembly.

3.1.4. Dynamic Overset Assembly Test

The capability to obtain dynamic overset assembly is tested on a moving cylinder in a channel test case. The same meshes and the algorithm settings as in Subsection 3.1.1. were used. Therefore, only the final overset mesh assemblies, in different time intervals, are shown in Figures 3.10 - 3.13.

Figure 3.10: Overset mesh assembly at $t = 0$ s.Figure 3.11: Overset mesh assembly at $t = 0.15$ s.

Figure 3.12: Overset mesh assembly at $t = 0.3s$.Figure 3.13: Overset mesh assembly at $t = 0.45s$.

3.2. Three-Levels Mesh of NACA 4412 Airfoil

In this section, the capability of the algorithm to obtain a static overset mesh assembly on a three-levels mesh is tested on the NACA 4412 test case.

As shown in Table 3.6, overset mesh system consists of three meshes, i.e. overset regions. The *backgroundMesh* region consists of 5 376 cells, *middleMesh* region consists of 8 064 cells and region *frontMesh* consists of 12 544 cells. Overall number of cells is 25 984 and all the cells are hexahedral.

Full view of the system is shown in Figure 3.14, while zoomed view is shown in Figure 3.15. Region *backgroundMesh* is coloured blue, region *middleMesh* is coloured green, while region *frontMesh* is coloured red.

Inter-mesh communication is established as following: region *backgroundMesh* communicates exclusively with region *middleMesh*, region *frontMesh* communicates exclusively with region *middleMesh*, while region *middleMesh* communicates both with *backgroundMesh* and *frontMesh* regions.

Table 3.6: NACA 4412 overset mesh system statistics.

Number of regions:	3		
Region name:	<i>backgroundMesh</i>	<i>middleMesh</i>	<i>frontMesh</i>
Number of cells per region:	5376	8064	12544
Overall number of cells:	25984		

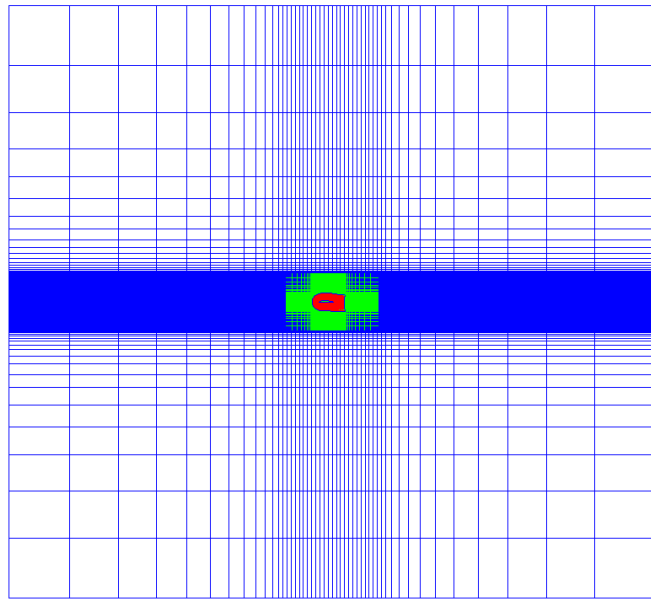


Figure 3.14: NACA 4412 overset meshes.

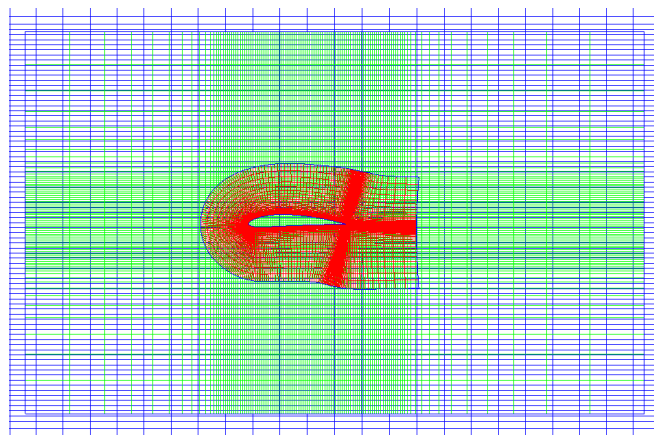


Figure 3.15: Zoomed view of NACA 4412 overset meshes.

In Table 3.7 overset fringe assembly settings are presented. The `adaptiveOverlap` algorithm is used together with `cellVolumes` DSF. The patch which determines holes is the airfoil wall. The initial patches are the outer patch of region *middleMesh* and the outer patch of the region *frontMesh*.

Table 3.7: Overset fringe assembly settings.

Region name:	<i>backgroundMesh</i>	<i>middleMesh</i>	<i>frontMesh</i>
Fringe type:	<code>adaptiveOverlap</code>	<code>adaptiveOverlap</code>	<code>adaptiveOverlap</code>
Donor regions:	<i>middleMesh</i>	<i>backgroundMesh</i> , <i>frontMesh</i>	<i>middleMesh</i>
<code>initPatchNames</code> :	/	<i>oversetMiddle</i>	<i>oversetFront</i>
<code>minLocalSuit</code> :	varied	varied	varied
<code>specifiedIterationsNumber</code> :	4	4	4
<code>additionalIterations</code> :	true	true	true
<code>orphanSuitability</code> :	-100	-100	-100
DSF:	<code>cellVolumes</code>	<code>cellVolumes</code>	<code>cellVolumes</code>
<code>holePatches</code> :	airfoil		

In the following Subsection 3.2.1., the iterative procedure steps are given, while in Subsection 3.2.2. only the final overset assembly is given.

Although during the iterative procedure all regions are interdependent, for clarity, the iterative procedure is described for all regions separately.

3.2.1. Test Case Setup 1

The first test case is performed with the minimum value for minimum local suitability parameter `minLocalSuit` = 0%. All the cells reach local suitability criterion. Therefore, unsuitable pairs consist of orphans only.

Iteration data is presented in the form of tables. The table row which represents the selected iteration for the final overset assembly is coloured green.

Iterative procedure for *backgroundMesh* region

The iterative procedure for region *backgroundMesh* was finished in 2 iterations because in iteration 2 there were no more unsuitable donor/acceptor pairs.

Iterative procedure steps are shown in Figures 3.16 and 3.17. Holes are coloured dark blue, acceptors from suitable donor/acceptor pairs are coloured red, while acceptors from unsuitable donor/acceptor pairs are coloured light blue. Overset types are shown in Figure 3.18.

Table 3.8: Iteration data for *backgroundMesh* region.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	2	-19.87	/
2	0	3.02	/

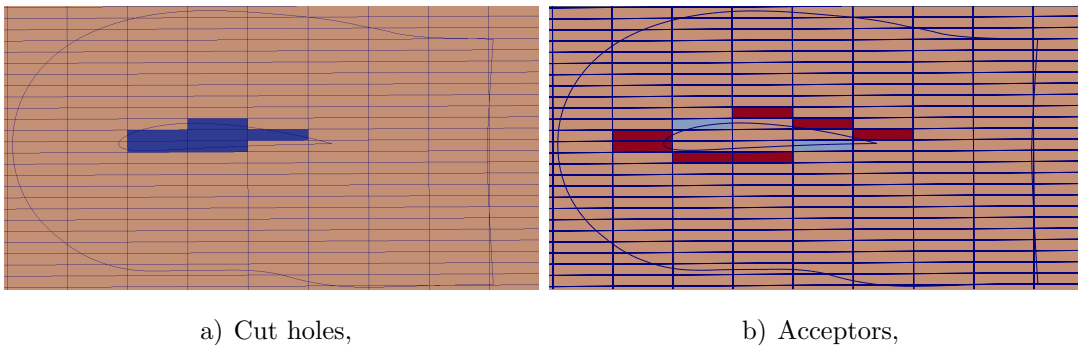


Figure 3.16: Iteration 1 for *backgroundMesh* region.

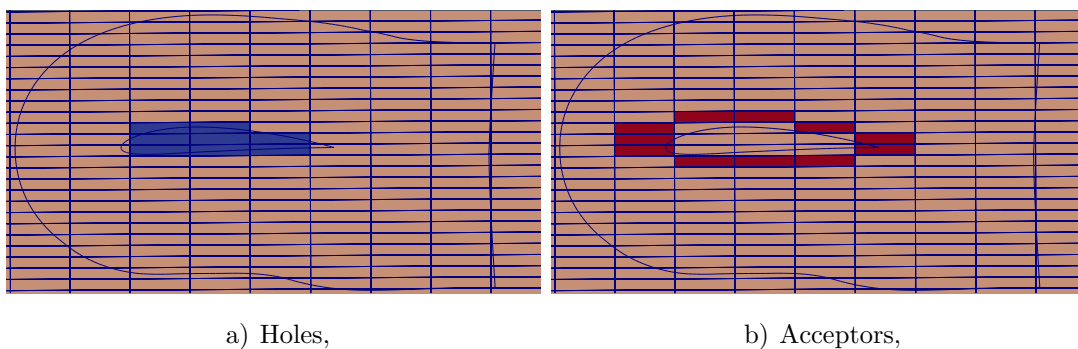


Figure 3.17: Iteration 2 for *backgroundMesh* region.

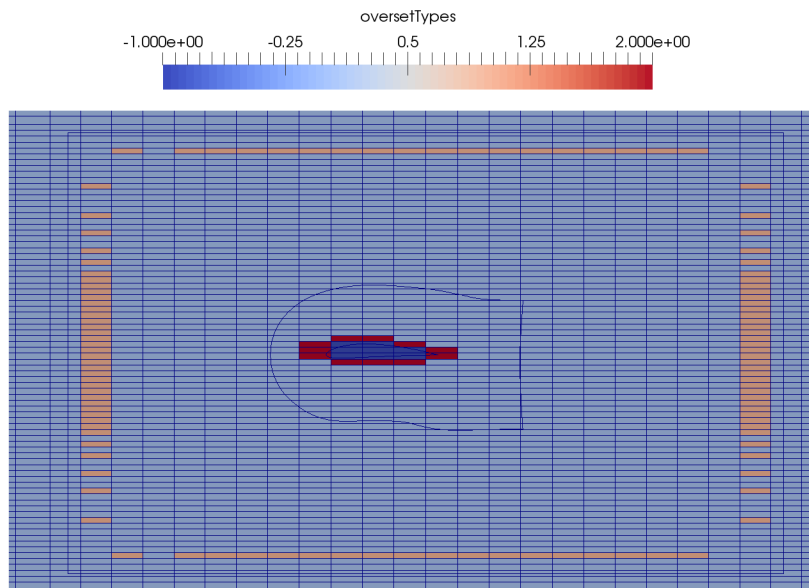


Figure 3.18: Overset types field for *backgroundMesh* region.

Iterative procedure for *middleMesh* region

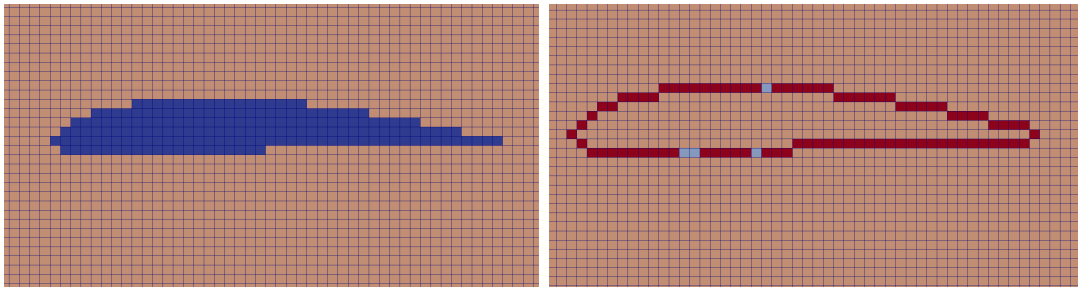
As for the region *backgroundMesh*, iterative procedure for region *middleMesh* was finished in 2 iterations because in iteration 2 unsuitable donor/acceptor pairs were not found.

Zoomed views of the iterative procedure steps are shown in Figures 3.19 and 3.20, while full views are shown in Figure 3.21.

After the filtering procedure was done, 4 acceptors were converted to holes. Overset types are shown in Figure 3.22.

Table 3.9: Iteration data for region *middleMesh*.

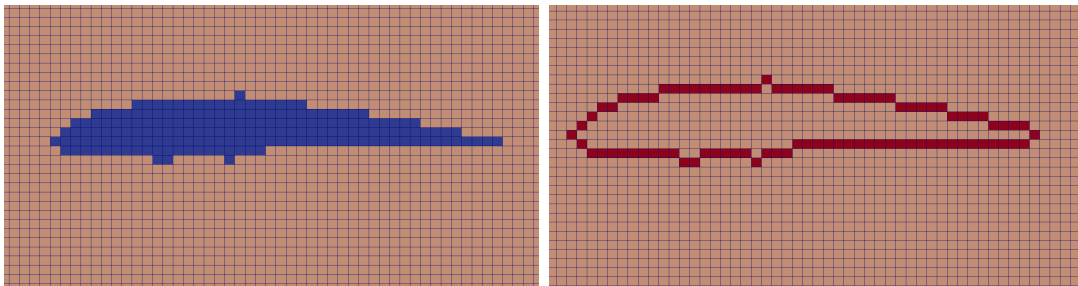
Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	4	44.03	/
2	0	44.94	/



a) Cut holes,

b) Acceptors,

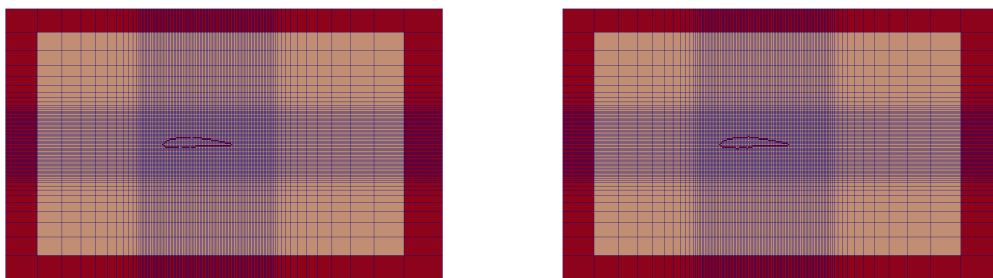
Figure 3.19: Iteration 1 for *middleMesh* region.



a) Holes,

b) Acceptors,

Figure 3.20: Iteration 2 for *middleMesh* region.



a) Acceptors (iteration 2),

b) Acceptors (iteration 3),

Figure 3.21: Iterations 2 and 3 for *middleMesh* region (full view).

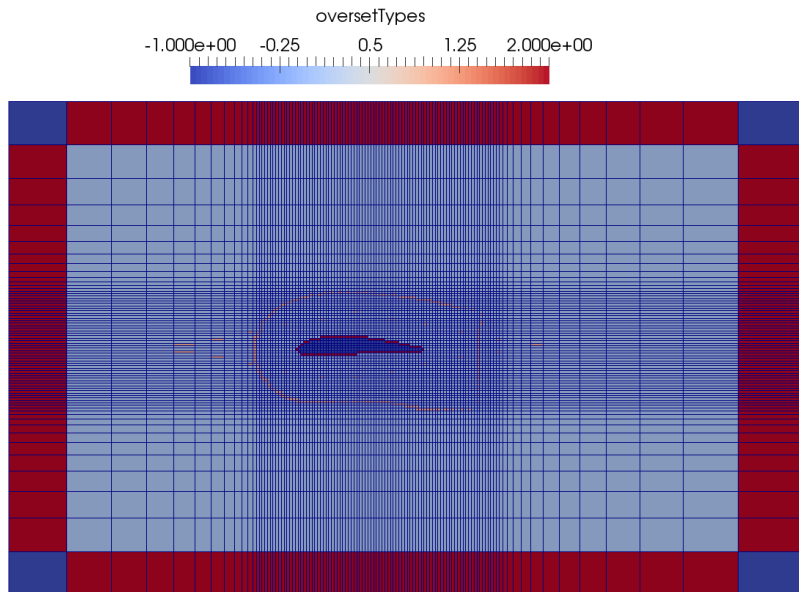


Figure 3.22: Overset types field for *middleMesh* region.

Iterative procedure for *frontMesh* region

The iterative procedure for region *frontMesh* was finished in 4 iterations. For overlap assembly, iteration with the highest average suitability was chosen, i.e. iteration 1. Additional iterations were not made, because the suitability slope was negative.

In Figures 3.23-3.26 the iterative procedure steps are shown. After the filtering procedure was done, 2 acceptors were converted to holes. Overset types field is shown in Figure 3.27, while the final overset mesh assembly is shown in Figure 3.28.

Table 3.10: Iteration data for *frontMesh* region.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	26	34.49	/
2	26	34.49	/
3	56	23.4	/
4	85	14.29	-7.16

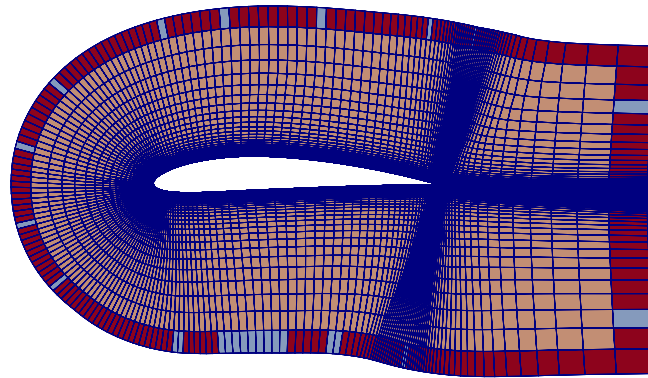
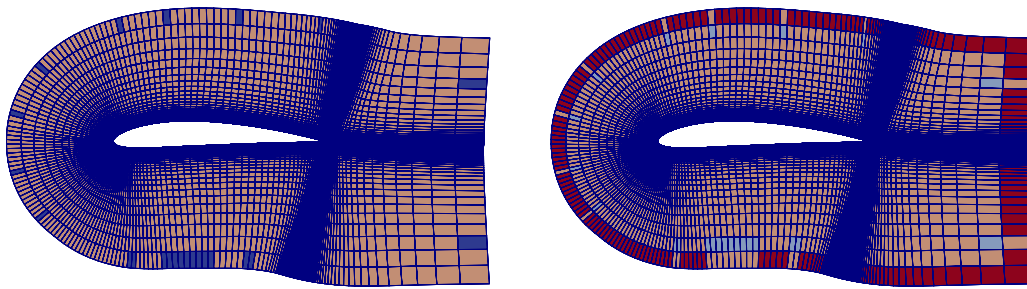


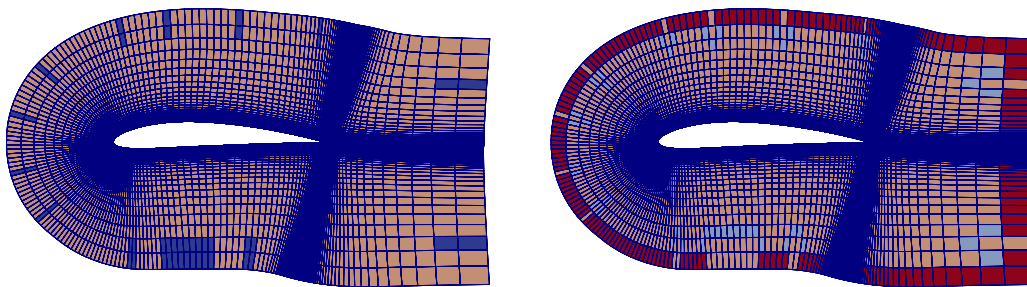
Figure 3.23: Iteration 1 for *frontMesh* region.



a) Holes,

b) Acceptors,

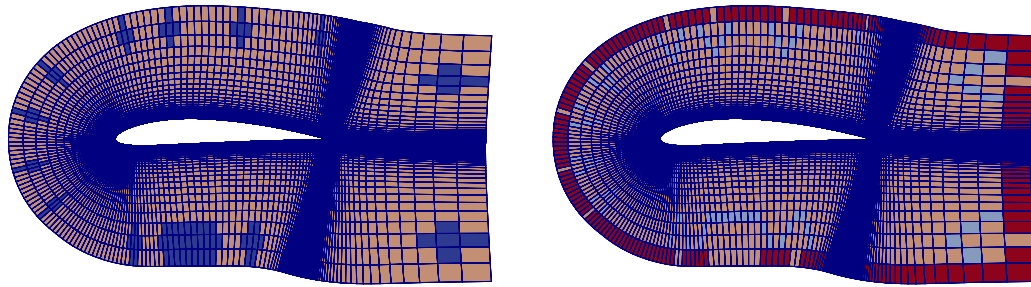
Figure 3.24: Iteration 2 for *frontMesh* region.



a) Holes,

b) Acceptors,

Figure 3.25: Iteration 3 for *frontMesh* region.



a) Holes,

b) Acceptors,

Figure 3.26: Iteration 4 for *frontMesh* region.

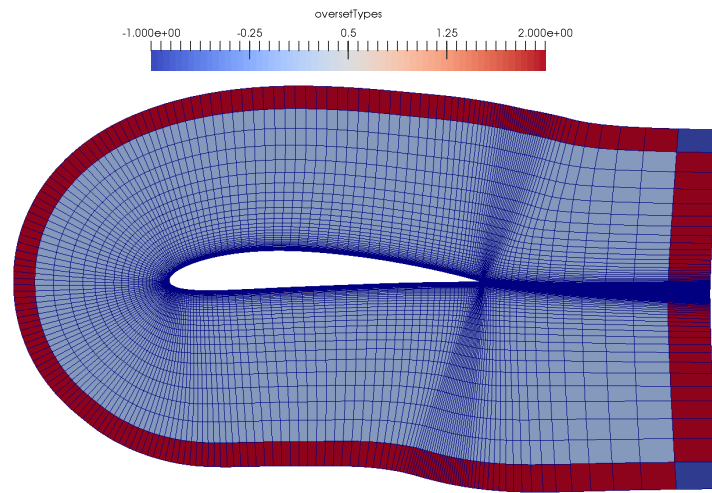


Figure 3.27: Overset types field for *frontMesh* region.

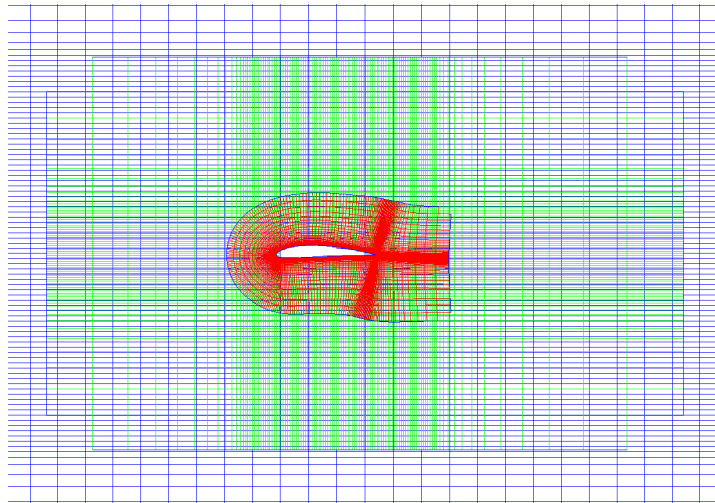


Figure 3.28: Final overset mesh assembly.

3.2.2. Test Case Setup 2

The second test case is performed with the maximum value for the minimum local suitability parameter $\text{minLocalSuit}=100\%$. In this case, all the cells are considered as unsuitable.

The iteration data is given in Tables 3.11-3.13. The number of unsuitable pairs is given. Here, the first number stands for orphans, while the second number stands for a cell which does not meet local suitability criterion.

For the region *backgroundMesh*, 1 additional iteration was made. The iterative procedure was finished in 5 iterations, due to the negative suitability slope. Chosen iteration is iteration 4.

For region *middleMesh* 4 iterations were made. The iterative procedure was finished after calculated suitability slope was negative. The chosen iteration is iteration 1. Four acceptors were converted to holes after filtering procedure was done.

For region *frontMesh* 5 iterations were made and chosen iteration is iteration 4. Two acceptors were converted to holes after the filtering procedure was done.

Overset types fields for all regions are given in Figures 3.29-3.31, while detail view of the final overset mesh assembly is given in Figure 3.32.

Table 3.11: Iteration data for *backgroundMesh* region.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	2 / 7	-19.87	/
2	0 / 13	3.34	/
3	0 / 17	4.55	/
4	0 / 21	6.67	8.08
5	25 / 0	-100	-30.78

Table 3.12: Iteration data for *middleMesh* region.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	4 / 450	44.1	/
2	3 / 447	40.73	/
3	2 / 444	38.62	/
4	7 / 435	35.8	-2.69

Table 3.13: Iteration data for *frontMesh* region.

Iteration	Number of unsuitable pairs	Average suitability [%]	Suitability slope [%/iter]
1	26/308	34.49	/
2	9/321	44.94	/
3	0/326	45.22	/
4	0/322	47.33	4.4
5	318/0	-100	-41.7

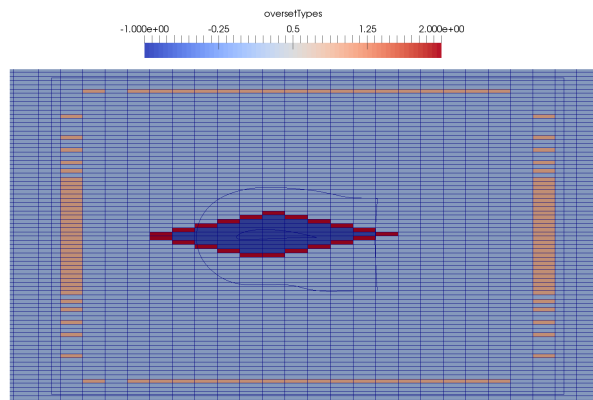


Figure 3.29: Overset types field for *backgroundMesh* region.

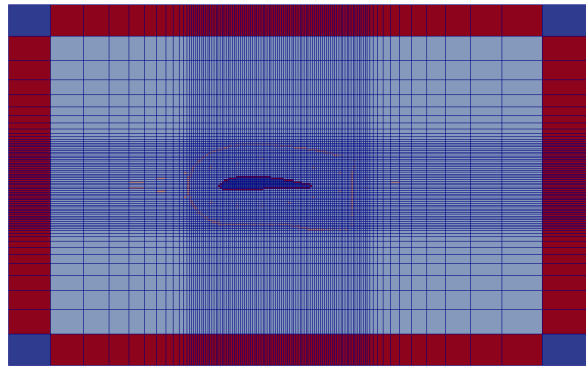


Figure 3.30: Overset types field for *middleMesh* region.

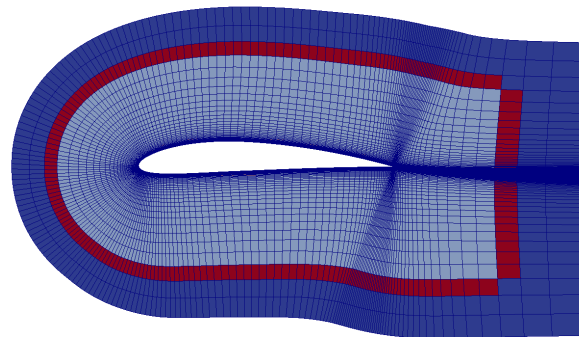


Figure 3.31: Overset types field for *frontMesh* region.

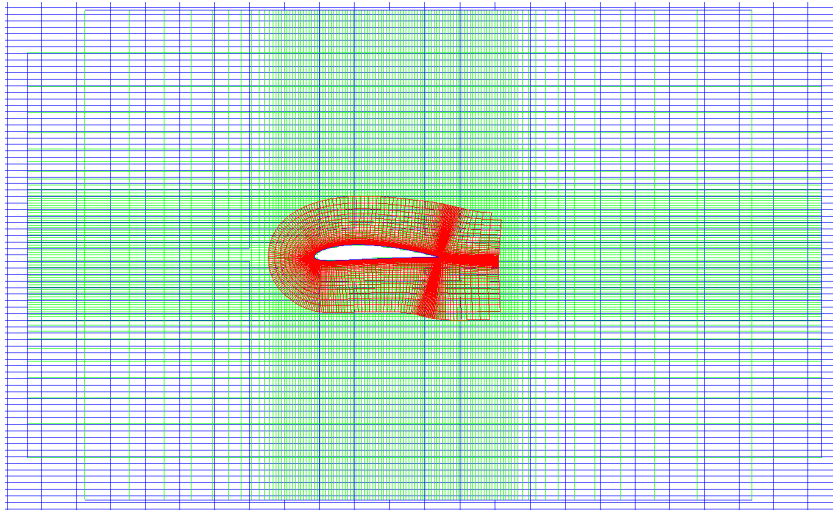


Figure 3.32: Detailed view of final overset mesh assembly.

3.3. Six-Levels Mesh of ONR Tumblehome Ship

The main goal of this test case is to test capability of the algorithm to produce static overset fringe assembly on an overset system which consists of 6 3D meshes. The system consists of 5 near-body meshes and 1 background mesh. Different views of the ONR Tumblehome ship geometry are shown in Figure 3.33.

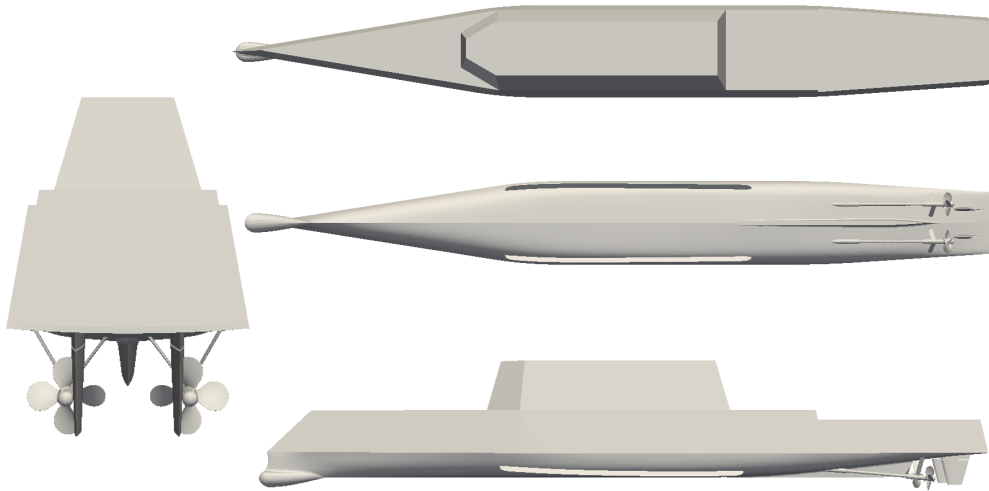


Figure 3.33: Different views of ONR Tumblehome ship.

Both propellers, as well as rudders, are spatially discretised with identical meshes. Overset mesh system statistics are presented in Table 3.14. The overall number of cells is 6 467 103.

Table 3.14: ONR Tumblehome overset mesh system statistics.

Number of regions:	6			
Region name(s):	<i>background</i>	<i>hull</i>	<i>propellerStarboard/ propellerPort</i>	<i>rudderStarboard/ rudderPort</i>
Number of cells per region:	1 383 300	1 876 649	1 178 626	424 951
Overall number of cells:	6 467 103			

For fringe assembly in all regions, `adaptiveOverlap` is employed along with `cellVolumes` DSF. For minimum local suitability parameter, maximum value is prescribed (`minLocalSuit`

=100%). Default values for other parameters are used. Inter-mesh communication is established as following:

- Region *background* communicates with region *hull*.
- Region *hull* communicates with regions *background*, *propellerStarboard*, *propellerPort*, *rudderStarboard* and *rudderPort*.
- Region *propellerStarboard* communicates with regions *rudderStarboard* and *hull*.
- Region *propellerPort* communicates with regions *rudderPort* and *hull*.
- Region *rudderStarboard* communicates with regions *propellerStarboard* and *hull*.
- Region *rudderPort* communicates with regions *propellerPort* and *hull*.

Walls of hull, propellers and rudders regions are defined as hole patches. Outer patches of propellers, rudders and hull regions are defined as initial patches. In Table 3.15, the chosen iterations, along with average suitability, are presented.

Table 3.15: Chosen iterations data.

Region:	Chosen iteration:	Average suitability [%]
<i>background</i>	4	25.84
<i>hull</i>	1	40.18
<i>propellerStarboard/ propellerPort</i>	1	15.18
<i>rudderStarboard/ rudderPort</i>	2	30.44

In Figures 3.34-3.37, overset types fields for different regions are shown. In Figure 3.35, the presence of few undesirable hole cells, which are surrounded by acceptors, is noticed. Those hole cells are the result of the error in the hole cutting procedure. After the hole cutting procedure was done, the addressing calculation procedure identified immediate neighbours of hole cells as acceptors. More detailed investigation of this test case should be conducted in future work.

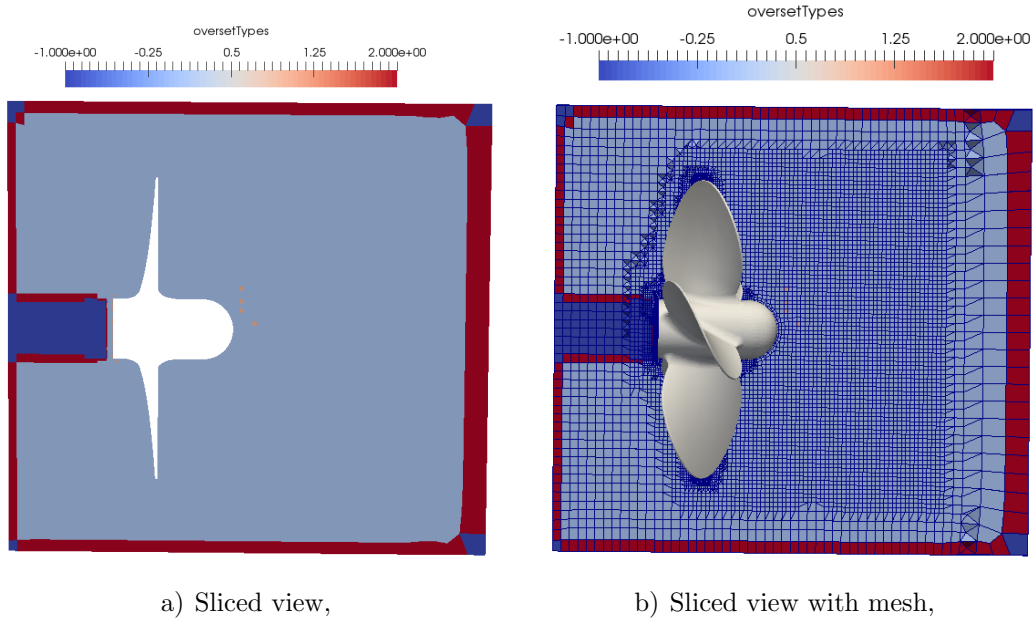


Figure 3.34: Overset types field for *propellerStarboard/propellerPort* regions.

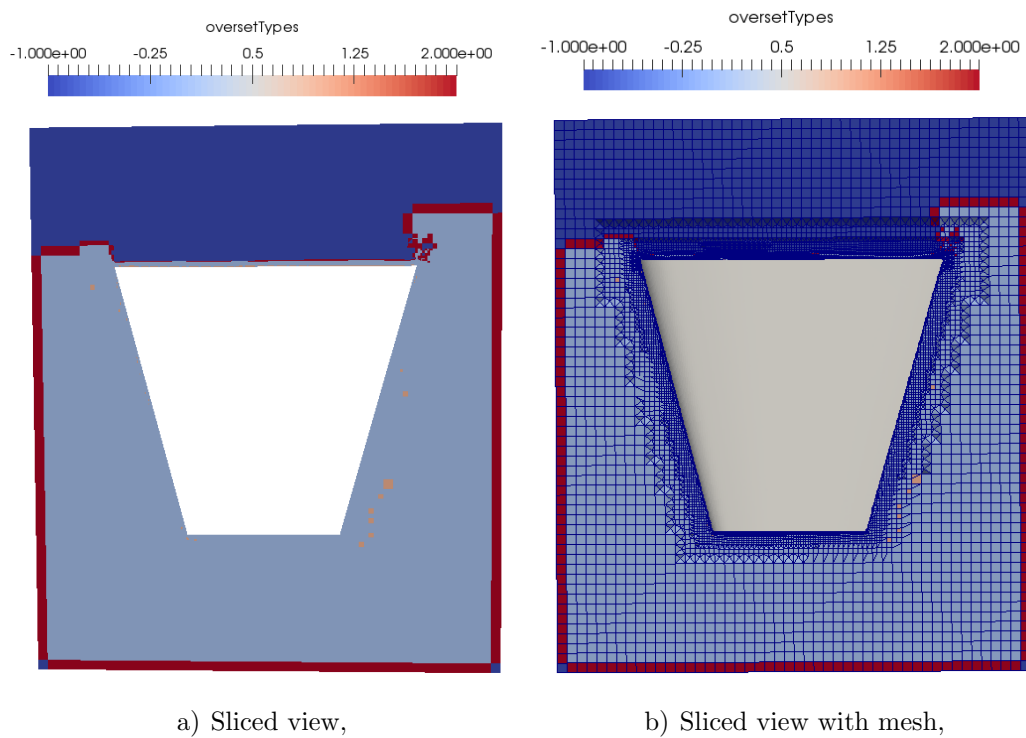


Figure 3.35: Overset types field for *rudderStarboard/rudderPort* regions.

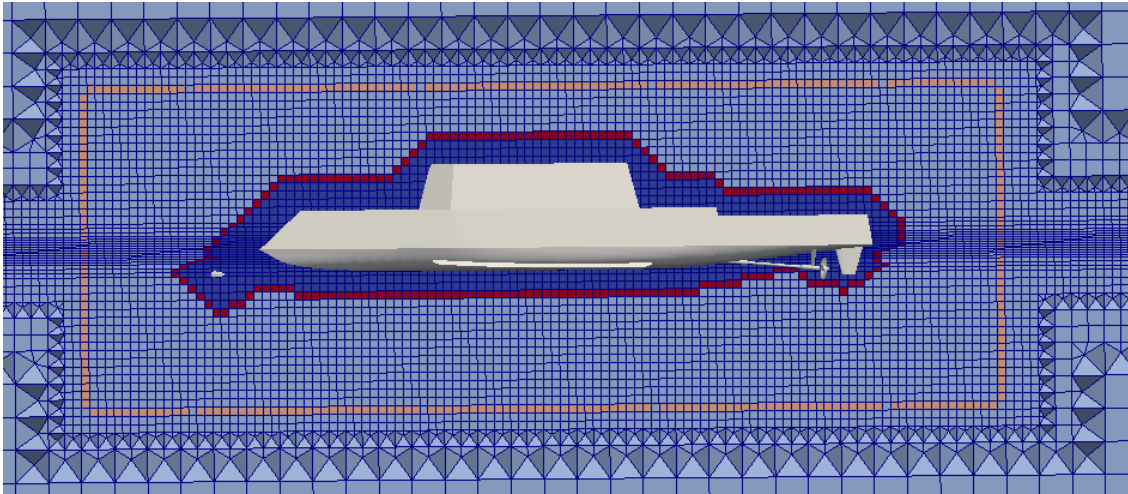


Figure 3.36: Overset types field for *background* region.

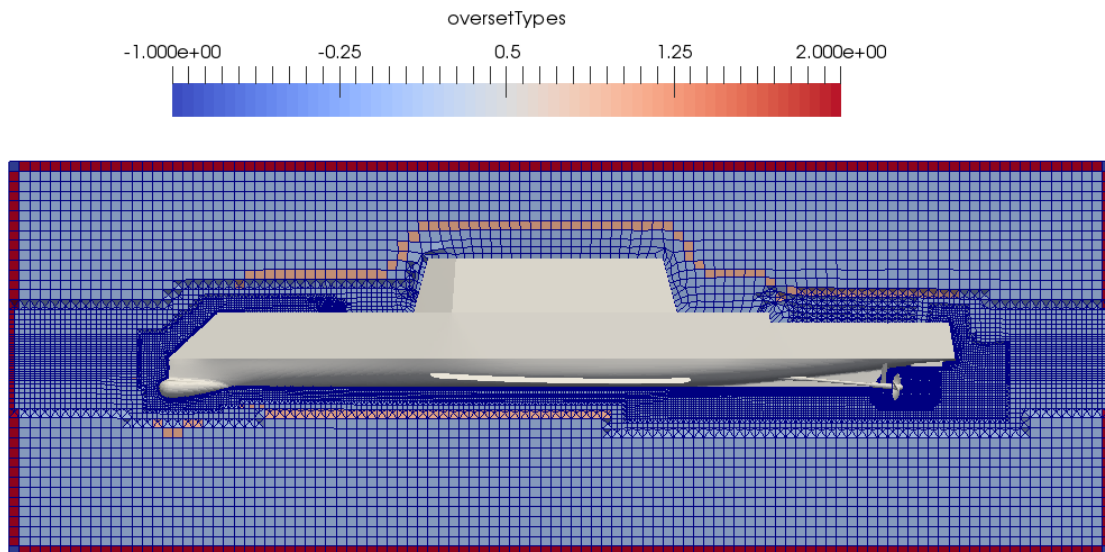


Figure 3.37: Overset types field for *hull* region.

3.4. Closure

In this chapter, testing of the newly developed algorithm as a pre-processing tool was performed on three test cases. Capabilities to produce both static and dynamic overset fringe assemblies are tested. The stopping criterion is tested by prescribing maximum

value for minimum local suitability parameter. In next chapter, the algorithm, as well as the *Naval Hydro* Pack, are tested and validated on side launching of a tanker CFD simulation.

4 Numerical Simulation of Side Launching of a 260 000 DWT Tanker by Tipping Table Principle

4.1. Introduction

In this chapter, the numerical simulation setup and the numerical results are presented in detail. The goal of the simulation is to test and validate the newly developed algorithm, as well as the *Naval Hydro Pack* which provides solvers and models for common applications for marine hydrodynamics [9]. Numerical results are compared with the experimental. The model test of the side launching of the tanker was ordered by Brodosplit, Shipbuilding Industry "Split" and was performed in *Maritime Research Institute Netherlands* (MARIN).

When the ship penetrates the water, a wave is generated. According to experimental data [3], the highest observed wave had a maximum amplitude of 2.2m, while the lowest maximum amplitude was 1m. Between those values, a wide scatter of maxima was observed. Measures should be taken for a generated wave of approximately 2m [3]. Due to diffraction effects, the generated wave can reach as much as twice the value observed at the wave probe [3].

Therefore, as a relevant physical quantity for testing and validation, the first generated wave amplitude is taken.

4.2. Numerical Case Setup

In this section, numerical simulation case setup is presented in detail. The main goal of the computational simulation is to compare experimental and numerical results for the first wave amplitude and phase in order to test and validate the *Naval Hydro* Pack.

The computational domain is described first, then the finite volume meshes with corresponding dimensions are displayed. Boundary conditions, overset mesh setup, simulation properties and prescribed motions are presented next. Finally, numerical results are given and discussed.

4.2.1. Computational Domain

The model test of the side launching of the tanker was performed in the Shallow Water Basin of MARIN by Feikeme *et. al* [3]. The harbour basin in front of the slipway was modelled according to Figure 4.1. The depth of the harbour was modelled by means of cement on the basin floor.

An existing wooden model in scale 1:68.74 was adapted for that purpose. The model of the ship was launched in a transverse direction from a horizontal position by the tipping table principle. 14 tests were carried out, where length between perpendiculars, displacement, the centre of gravity above the base, metacentric height of a ship, water depth, etc. were varied.

The prototype, which can be considered an elastic girder, is supported by 12 tipping tables. Thus the supporting forces are evenly distributed over the tipping tables. The model, however, is very rigid and the support of the model is statically indeterminate when more than two tipping tables are used. Under the assumption of a constant friction coefficient, it can be shown that the number and position of the tipping tables do not influence the dynamics of the launching even if external forces are taken into account. It was therefore decided to use two tipping tables [3].

The results of the measurements have been scaled up to full-size values according to Froude's law. The results are given as time traces in figures [3].

The main data of the water basin with bottom topography and the ship model for the test case No. 5874 are shown in Figure 4.1. According to the coordinate system shown in Figure 4.2, the deepest point of the Water Basin is at -17m, while the shallowest point is at -9m.

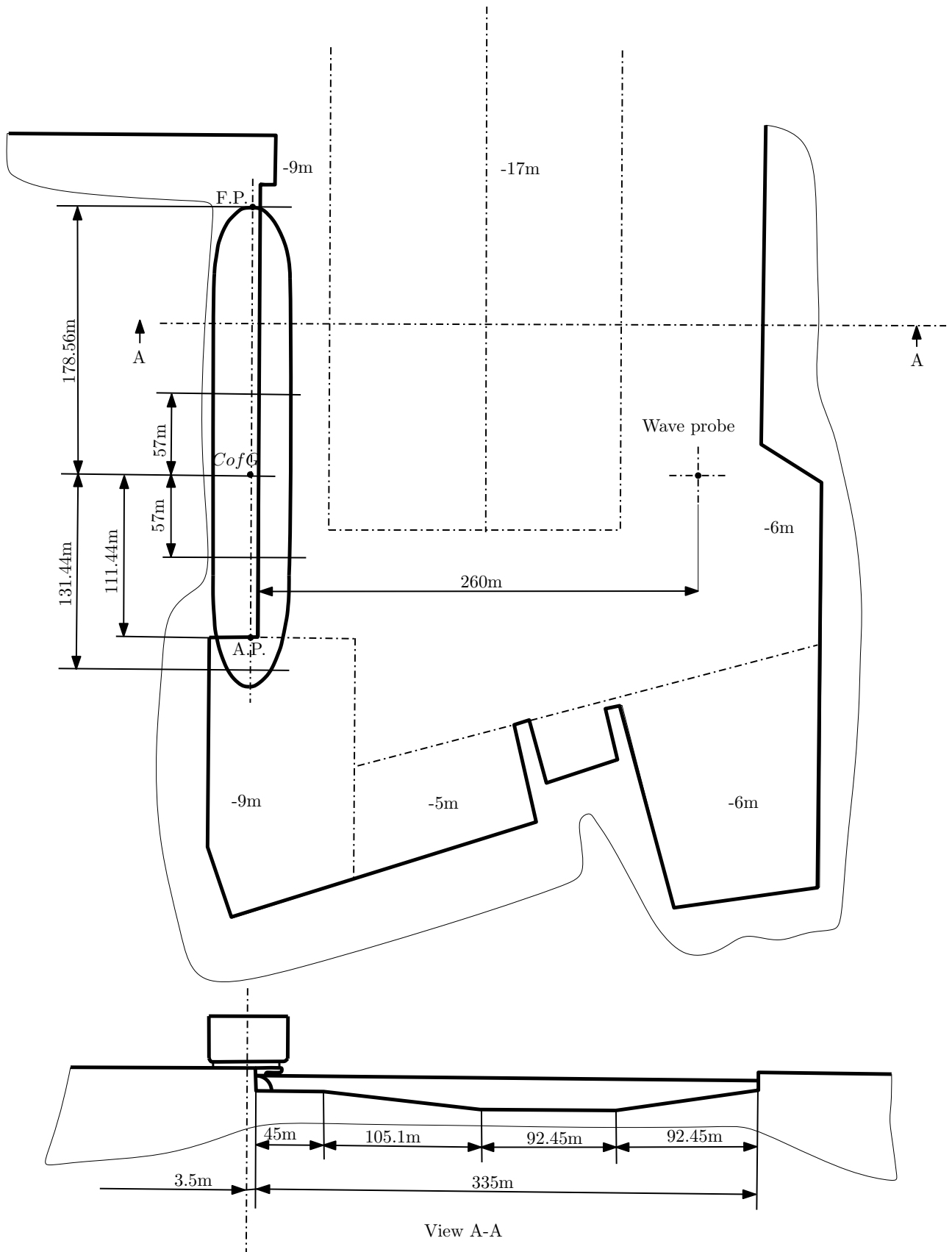


Figure 4.1: Water basin with ship model on tipping table [3].

According to the water basin dimensions (Figure 4.1), the 2D computational domain was designed. The computational domain is spatially discretised with two oversight meshes. The first mesh is called *backgroundMesh* and the second is called *boatMesh*. Meshes arrangement and free surface position at the initial moment can be seen in Figure 4.2.



Figure 4.2: Meshes arrangement and free surface position at initial moment.

The *backgroundMesh* mesh dimensions can be seen in Figure 4.3. Overall mesh length is 396 m, while overall mesh height is 63 m. Initial free surface position is set to $(0, 0)$ in accordance with $x - y$ coordinate system (Figure 4.4).

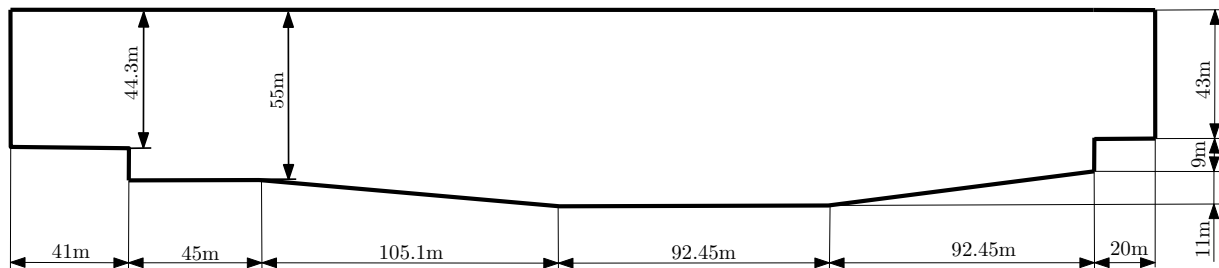


Figure 4.3: Dimensions of *backgroundMesh* region.

The *boatMesh* mesh dimensions are shown in Figure 4.4. Overall mesh length is 64 m, while overall mesh height is 41.05 m.

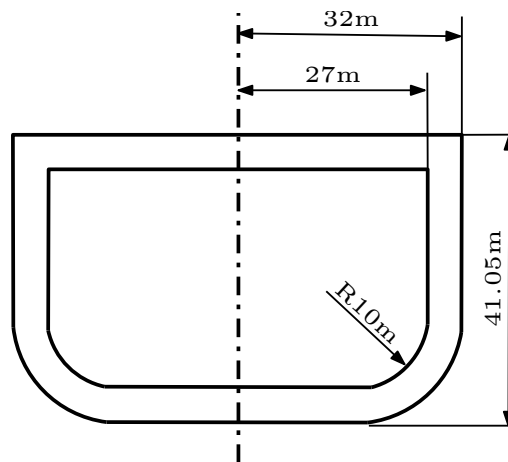


Figure 4.4: Dimensions of *boatMesh* region.

4.2.2. Finite Volume Meshes

The *boatMesh* and the *backgroundMesh* finite volume meshes are generated using *Pointwise* software. The meshes are block-structured. The *backgroundMesh* consists of 52 724 cells, while the *boatMesh* mesh consists of 21 174 cells. All the cells are hexahedral. The overset mesh system thus consists of 73 898 cells.

The *backgroundMesh* mesh is made with 20 elements per wave height. The full view of the *backgroundMesh* can be seen in Figure 4.5, while the detailed view of the mesh which shows the mesh grading around initial free surface position can be seen in Figure 4.6.

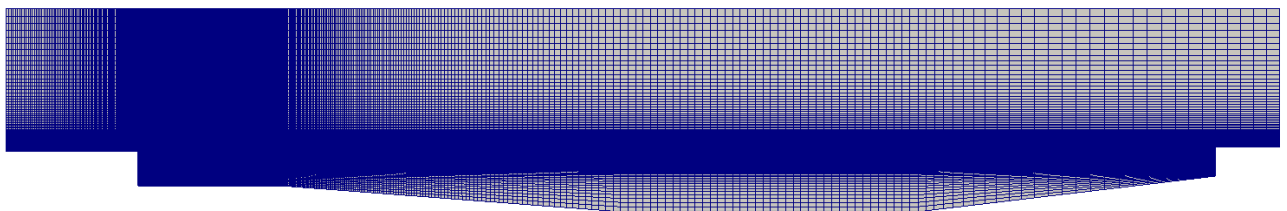


Figure 4.5: Full view of the *backgroundMesh* region.

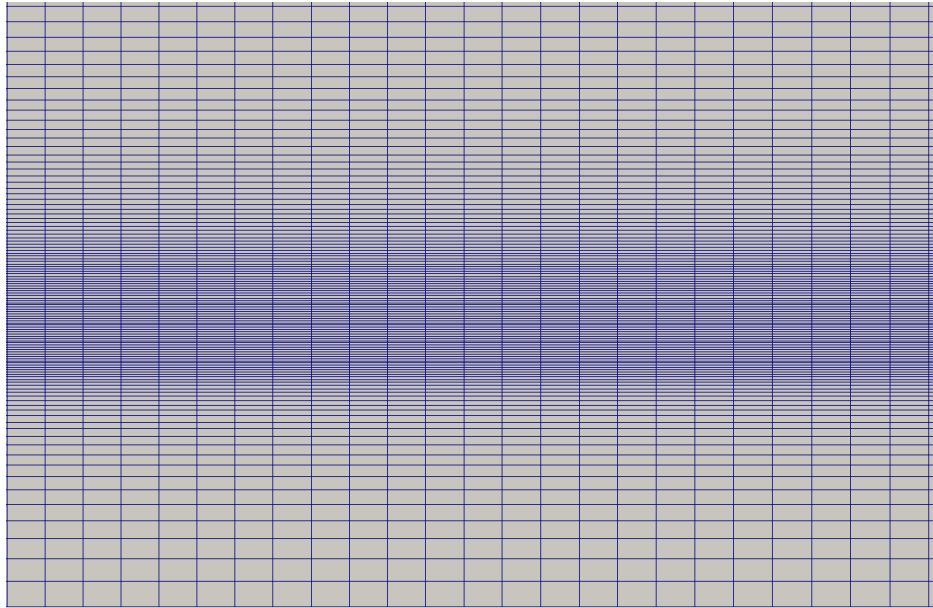


Figure 4.6: Detailed view of the *backgroundMesh* region around initial free surface position.

The *boatMesh* mesh is shown in Figure 4.7. To ensure a good quality overlap assembly, the mesh is generated in a way that the bottom and side blocks consist of 40 layers of cells, while upper blocks, which are not relevant for wave elevation, consist of 18 layers of cells. The width of the mesh is 5 m.

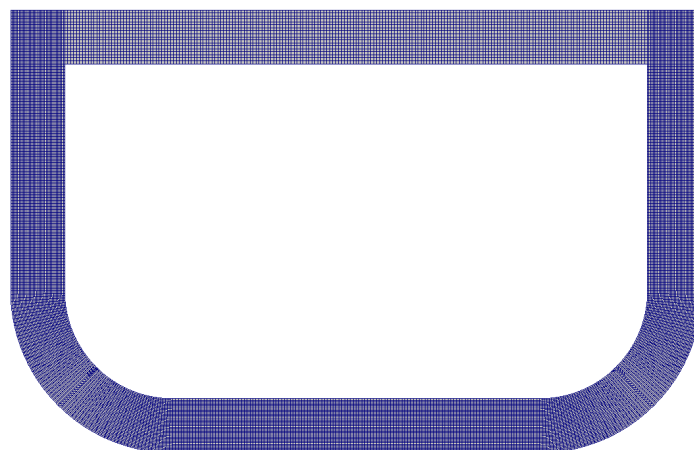


Figure 4.7: Full view of the *boatMesh* region.

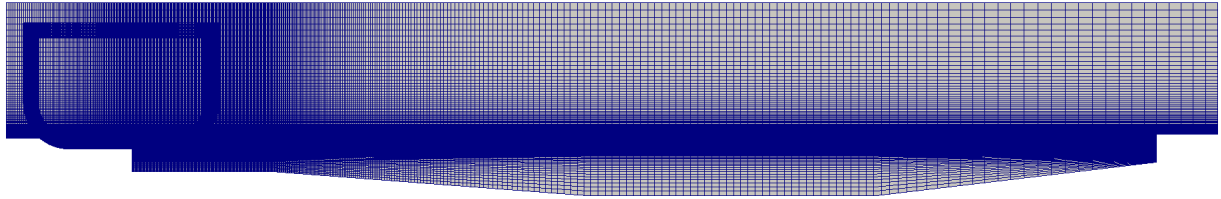


Figure 4.8: Combined meshes at initial moment of side launching.

4.2.3. Boundary Conditions

Patch IDs are marked in Figure 4.9. Prescribed boundary conditions for velocity, phase fraction and dynamic pressure fields along with patch IDs and names are given in Table 4.1. In addition to the boundary conditions given in Table 4.1, overset boundary conditions for every field are prescribed. The overset boundary conditions are used for overset interpolation, and to prescribe hole cell values for certain fields.

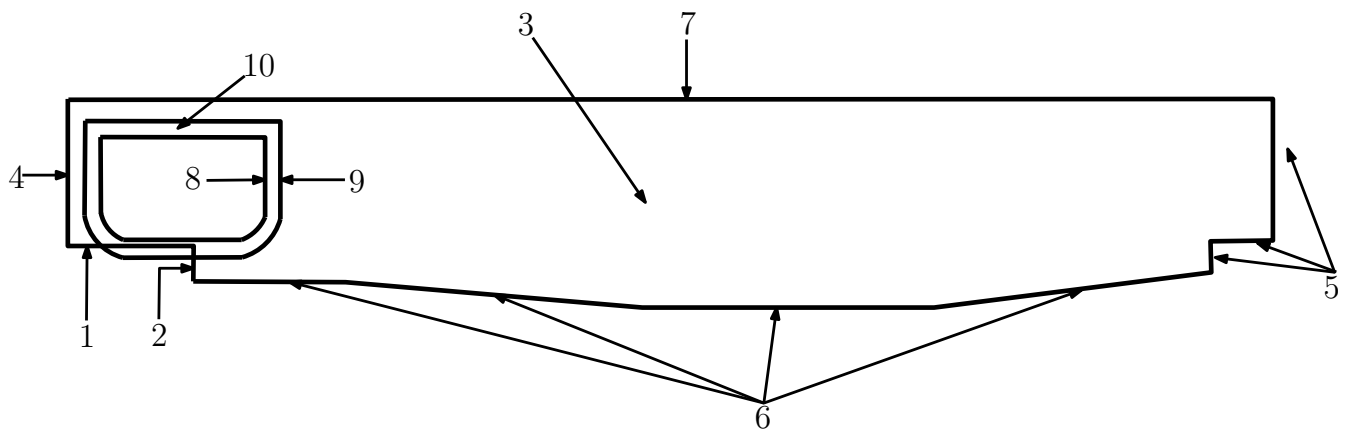


Figure 4.9: Computational domain with patch IDs marked.

Table 4.1: Prescribed boundary conditions.

Patch ID	Patch name	Velocity	Phase fraction	Dynamic pressure
1	coastHorizontal	fixedValue	zeroGradient	zeroGradient
2	coastVertical	fixedValue	zeroGradient	zeroGradient
3	frontAndBack	empty	empty	empty
4	left	inletOutlet	inletOutlet	zeroGradient
5	right	fixedValue	zeroGradient	zeroGradient
6	seabed	fixedValue	zeroGradient	zeroGradient
7	top	pressureInletOutletVelocity	inletOutlet	fixedValue
8	boat	movingWallVelocity	zeroGradient	zeroGradient
9	boatOuter	emptyOverset	emptyOverset	emptyOverset
10	frontAndBackBoat	empty	empty	empty

4.2.4. Overset Mesh Setup

The computational domain is spatially discretized with two overset meshes, which indicates that the overset mesh system consists of two overset regions. The first overset region is called *boatMesh* and the second is called *backgroundMesh*. Donor region for the *boatMesh* is *backgroundMesh*, while donor region for the *backgroundMesh* is *boatMesh*. For overset mesh assembly `adaptiveOverlap` with default parameters is employed. Patches which determine holes are presented in Table 4.2.

Table 4.2: Hole patches.

Patch ID	Patch name
1	coastHorizontal
2	coastVertical
4	left
5	right
6	seabed
7	top
8	boat

Chosen interpolation scheme for all fields is `inverseDistance`.

4.2.5. Prescribed Motions

Numerical simulations are performed with both theoretical and experimental motions, and wave elevations are compared.

Experimental data is extracted from diagrams given in [3]. The linear motions were measured for aft and fore of the ship. As 2D computations are performed, the motions are reduced to the centre of gravity of a ship (CofG) using inverse distance weighting (IDW). A general form of finding an interpolated value u at a given point \mathbf{x} based on samples $u_i = u(\mathbf{x}_i)$ for $i = 1, 2, \dots, N$ using IDW is an interpolation function [20]

$$u(\mathbf{x}) = \begin{cases} \frac{\sum_{i=1}^N \omega_i(\mathbf{x})u_i}{\sum_{i=1}^N \omega_i(\mathbf{x})}, & \text{if } d(\mathbf{x}, \mathbf{x}_i) \neq 0 \text{ for all } i \\ u_i, & \text{if } d(\mathbf{x}, \mathbf{x}_i) = 0 \text{ for some } i \end{cases} \quad (4.1)$$

where

$$\omega_i = \frac{1}{d(\mathbf{x}, \mathbf{x}_i)^p} \quad (4.2)$$

is a simple IDW weighting function where \mathbf{x} denotes an interpolated point, \mathbf{x}_i is an interpolation point, d is given distance from the interpolated to the interpolation point, N is the total number of interpolating points used in interpolation and p is a positive real number, called the power parameter. In this case, used power parameter is $p = 1$. Here weight decreases as distance increases from the interpolated points.

Distance from aft to CofG is $A = 178.56$ m, while distance from fore to CofG is $F = 131.44$ m (Figure 4.1). According to Eq. 4.1 weight factors of aft and fore motions are

$$weightA = 0.424 \quad (4.3)$$

$$weightF = 0.576 \quad (4.4)$$

The extracted and interpolated motions are shown in Figure 4.10 and 4.11. The extracted motions are coloured blue (fore motion) and red (aft motion), while interpolated, i.e. prescribed motions are coloured green.

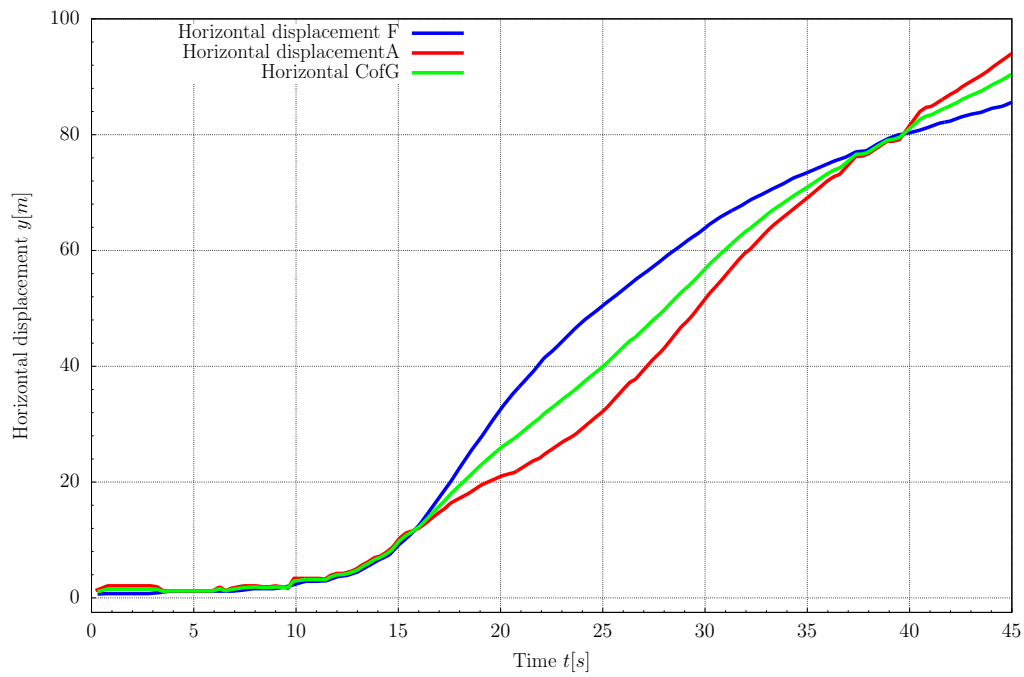


Figure 4.10: Horizontal displacement (experimental data).

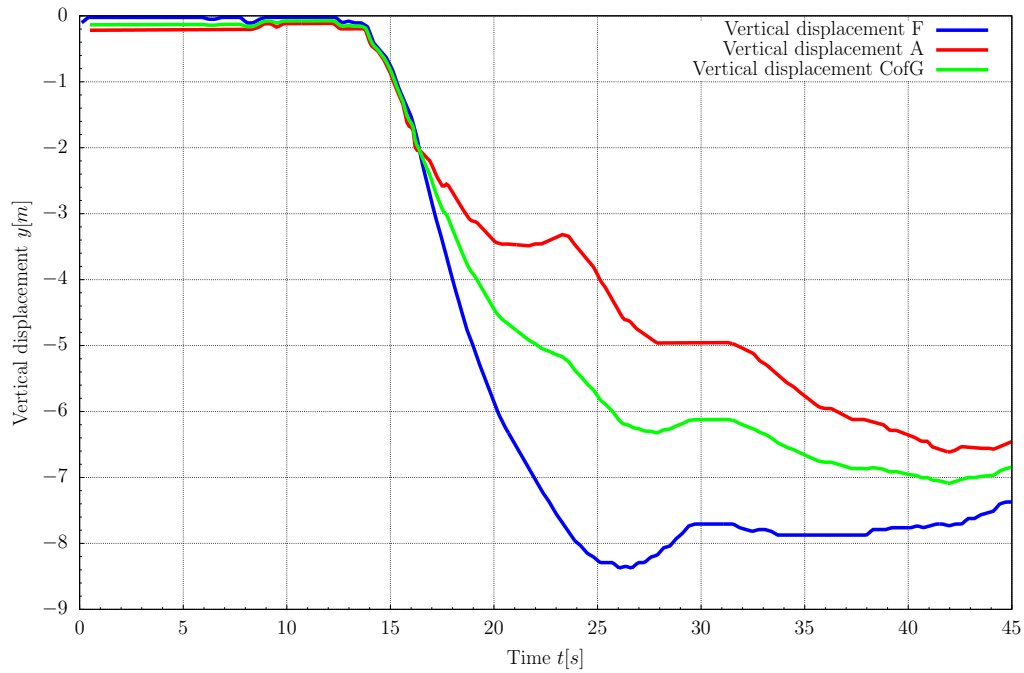


Figure 4.11: Vertical displacement (experimental data).

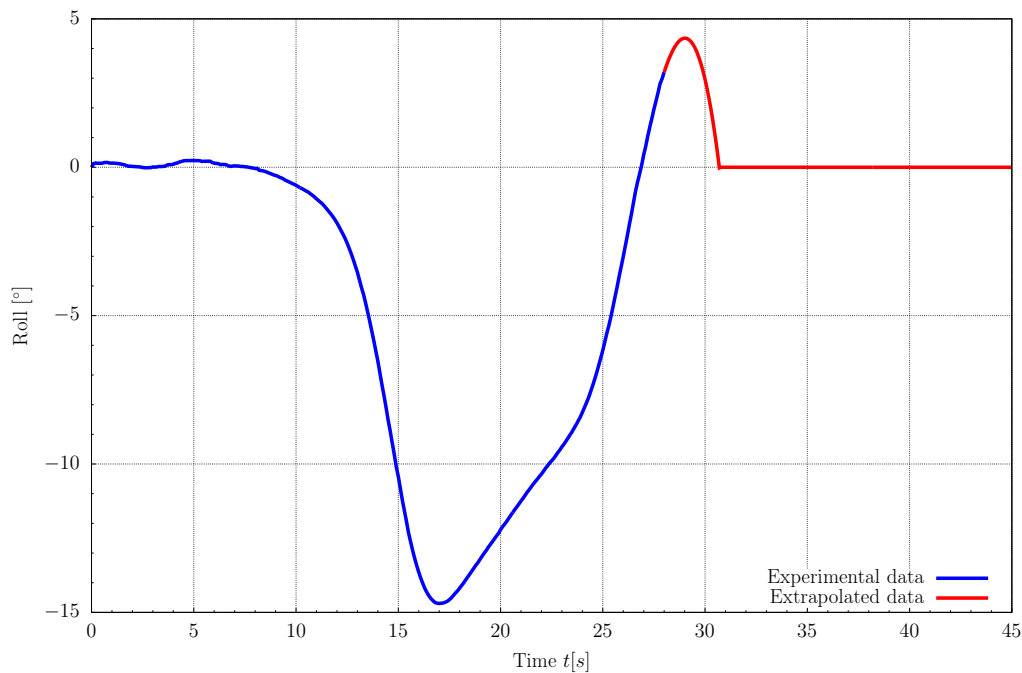


Figure 4.12: Roll angle (experimental data).

The roll angle is measured only for the first 28 s and is extrapolated to capture the first wave at the wave probe location. The roll angle is extrapolated in such a way that the ship returns to a neutral position and stays therein.

In Figures 4.2.5. and 4.2.5. prescribed theoretical motions are shown. The theoretical motions are obtained by solving equations of motion presented in Section 1.3.1. Nonlinear differential equations of motion are solved by the finite difference method using the software named LATUP (LAunching by TUrning Pads) [5].

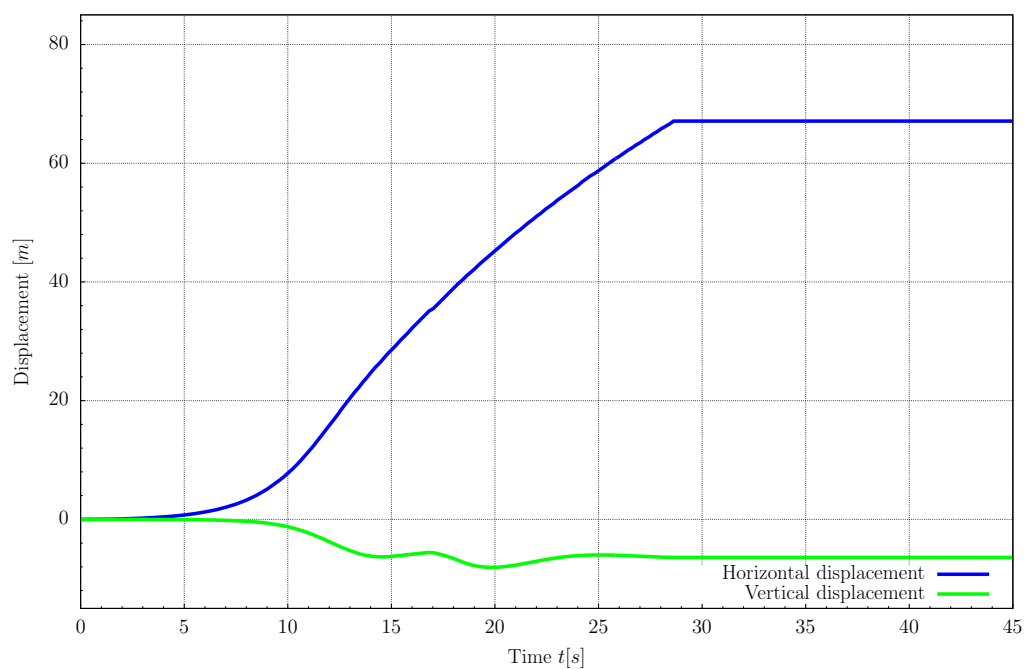


Figure 4.13: Prescribed horizontal and vertical displacement.

Theoretical motions are available for the first 5 launching phases, i.e. until final phase, where ship freely floats performing complex motion. To capture first wave elevation, motions are extrapolated.

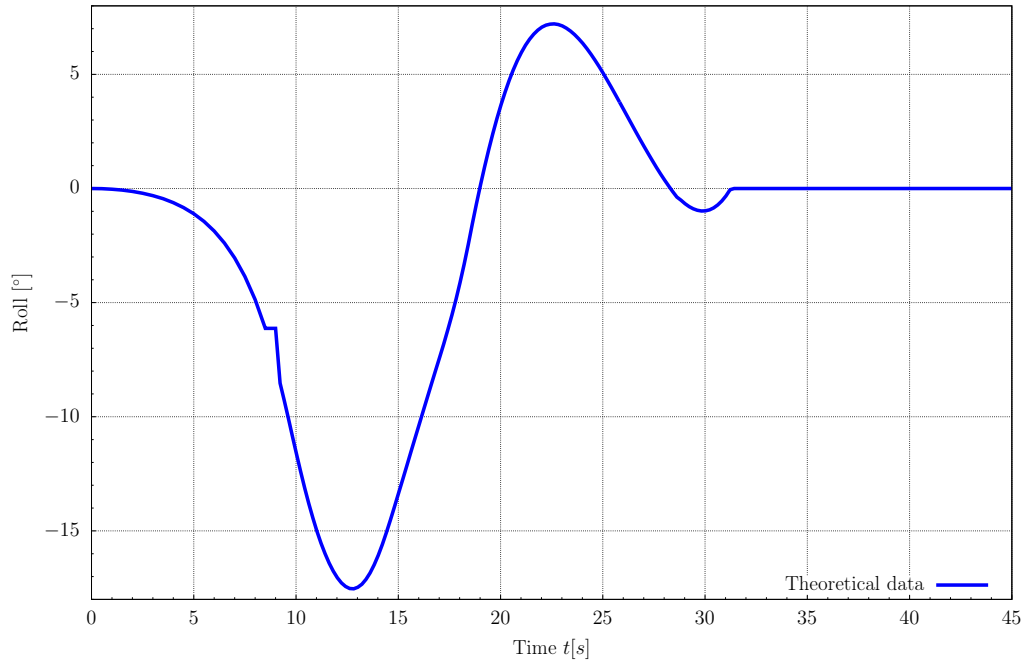


Figure 4.14: Prescribed roll motion.

To prescribe mesh motion with no topology change in mesh `oversetSolidBodyMotionFvMesh` motion type is used.

4.2.6. Simulation Properties

Two phases used in this case are water and air. For tracking and locating the free surface (or fluid-fluid interface), the volume of fluid method (VOF) is used. Density of the water is $\rho_w = 998 \text{ kg/m}^3$, while its kinematic viscosity is $\nu_w = 1.05 \cdot 10^{-6} \text{ m}^2/\text{s}$. Density of air in this case is set to $\rho_a = 1 \text{ kg/m}^3$, while its kinematic viscosity is set to $1.48 \cdot 10^{-5} \text{ m}^2/\text{s}$. Surface tension coefficient is $\sigma = 0 \text{ kg/s}^2$ because surface tension effects are considered negligible for large-scale flows [21]. The flow of both phases is laminar. Waves are generated exclusively due to ship's penetration in water, radiation and diffraction effects. Therefore `noWave` wave theory was used which means that at initial moment water in the basin is calm and waves are generated only due to the motion of the ship.

4.2.7. Results and Discussion

In Figures 4.15-4.20 phase fraction fields at different moments are given. The water phase is coloured blue, where a darker shade of blue stands for deeper points of a computational domain.

Surface elevation at the probe location is given in Figure 4.21, where CFD experimental data stands for calculated surface elevation using CFD with measured motions prescribed, EFD stands for measured surface elevation, and CFD theoretical data stands for calculated surface elevation using CFD with theoretical motions prescribed.



Figure 4.15: Phase fraction field at $t = 0$ s.



Figure 4.16: Phase fraction field at $t = 15$ s.



Figure 4.17: Phase fraction field at $t = 19$ s.



Figure 4.18: Phase fraction field at $t = 21.3$ s.



Figure 4.19: Phase fraction field at $t = 28.9$ s.



Figure 4.20: Phase fraction field at $t = 35.9$ s.

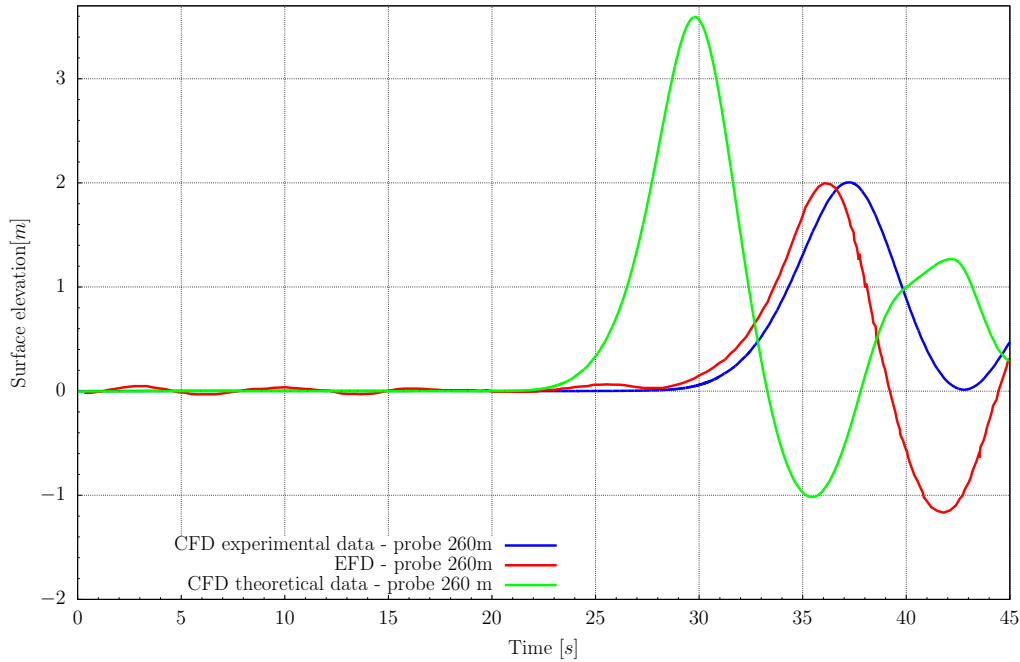


Figure 4.21: Surface elevation.

The maximum measured wave amplitude is 1.997 m, and it is measured at $t = 36.04$ s, which corresponds to wave amplitude from experimental test case No. 5874. Maximum calculated wave amplitude is 2.004 m at $t = 37.22$ s. Here the relative error for wave amplitude is 0.36%. The phase difference of 1.17 s between experimental and numerical data is noticed where relative error is 3.16%. The discrepancies are acceptable.

The second wave amplitude is underestimated which should not be considered as relevant because the prescribed roll motion is extrapolated for $t > 28$ s.

Observable difference between CFD results with experimental motions prescribed and CFD results with theoretical motions is noticed. It occurs because theoretical motions overestimate vertical displacement of the ship, i.e. according to theoretical motions, the ship penetrates deeper in the water than it is a case with experimental motions.

Also, when ship penetrates the water, the free surface is raised for about 0.5 m due to nature of 2D simulation where boundary conditions do not allow fluid flow through *frontAndBack* and *frontAndBackBoat* patches. This disturbance propagates after the first wave amplitude is captured. This indicates that 2D simulation is an appropriate

tool only for capturing the first wave elevation. Also, the maximum measured yaw angle for the test case No. 5874 is 6.5° , which indicates that the ship performs 3D motion. 3D simulation is not performed, because not all water basin dimensions were given, which would also affect the results due to 3D effects.

Surface elevation is extremely sensitive to prescribed motions. The prescribed motions of aft and fore are extracted from diagrams and reduced to CofG of a ship using inverse distance weighting. Extracting and interpolation represent a significant error source, which affects final surface elevation curve.

4.2.8. Closure

This section covered side launching theory, computational simulation setup and results. The main goal of the computational simulation was to compare experimental and numerical results for the first wave amplitude and phase.

Relative error between measured and calculated wave amplitudes is 0.36%, while relative error between their phases is 3.16%. It is concluded that numerical simulation captures side launching physics with high accuracy. It is concluded that 2D simulation is an appropriate tool only for capturing first wave elevation, due to 3D effects which occur during side launching of a ship. Furthermore, comparing simulations with experimental and theoretical motions shows significant sensitivity of the wave amplitude to the imposed motions.

5 Conclusion

Implementation and testing of automatic fringe assembly algorithm for overset meshes are conducted in this work, as well as validation of the *Naval Hydro Pack* which provides solvers and models for marine hydrodynamics. The algorithm is implemented in `foam-extend` C++ software environment. `foam-extend` is an open-source software specialised for continuum mechanics simulations using Finite Volume Method (FVM).

The newly developed algorithm determines fringe layer robustly and reliably. It tracks iteration history and chooses the iteration with the highest average suitability for the final overlap assembly, i.e. it has fall-back mechanism implemented. The algorithm uses modified DSFs which provide a realistic insight into cells similarity. Depending on the average suitability slope, it makes additional iterations or stops the iterative procedure.

Definition of the orphan is changed. Therefore, the master donor is an eligible cell whose cell centre is the closest to acceptor's cell centre, which means that the final overlap assembly will always be found, even when mesh resolutions in the overlapping region do not match well. Orphan suitability is not calculated in a standard manner using DSFs - negative value is prescribed. By prescribing negative suitability to an orphan, average suitability is being degraded. That reduces a possibility that fringe assembly with an orphan present is being chosen for the final overset assembly.

Stopping criteria are improved. When the algorithm propagates till region's boundary, lack of eligible cells, i.e. orphans presence, degrades average suitability which finally results in negative suitability slope stopping the iterative procedure for current region.

The algorithm was tested on three test cases, as a pre-processing tool and on one

test case with fluid flow present. The pre-processing test cases are a two-levels mesh of a cylinder in a channel, three-levels mesh of NACA 4412 airfoil and six-levels mesh of ONR Tumblehome ship. The algorithm parameters were varied, as well as grading of a cylinder mesh.

Pre-processing test cases show that the algorithm is capable of producing valid static and dynamic overset fringe assemblies, even with unreasonably high user-prescribed value for minimum local suitability. It was shown that prescribed orphan suitability of -7% was enough to stop the iterative procedure in the cylinder in channel test case with standard grading. Undesirable hole cells in ONR Tumblehome fringe assembly, as a result of the error in hole cutting procedure, were noticed. More detailed investigation of the ONR Tumblehome test case should be conducted in future work.

The last test case simulates side launching of a tanker. In comparison with experimental data, a difference in the first wave amplitude of 0.007 m is noticed (relative error is 0.36%), as well as the phase difference of 1.17 s (relative error is 3.16%). Results show good agreement, providing proof of the validity of the newly developed algorithm and the *Naval Hydro Pack*.

Observable difference between CFD results with experimental motions prescribed and CFD with theoretical motions prescribed was noticed. It occurs because theoretical motions overestimate vertical displacement of a ship. Therefore, it was concluded that surface elevation is extremely sensitive to prescribed motions.

Further, it was concluded that a 2D simulation is an appropriate tool only for capturing the first wave elevation, due to nature of 2D simulation where 2D boundary conditions do not allow flow through front and back patches, which causes the free surface raise of about 0.5 m when ship penetrates the water. This disturbance propagates after the first wave amplitude is captured. Also, 3D effects, i.e. radiation and diffraction appear.

Bibliography

- [1] Ivo Senjanović. Numerical simulation and model test of side launching of a 260000 DWT tanker by tipping table principle. *International shipbuilding progress*, 47(450):161–181, 2000.
- [2] Ivo Senjanović. Launching of ships and floating structures from horizontal berth by tipping table. *Journal of ship production*, 14(4):265–276, 1998.
- [3] G.J. Feikeme, H. Luisman, and J.A. Pinkster. *Report No. 010001-1-BT: Launching tests for a 260 000 DWT ship*. Maritime Research Institute Netherlands (MARIN), 1989.
- [4] Ivo Senjanović. Numerical and experimental investigation of side launching of tanker by tipping table principle. *Brodogradnja*, 46(3):264–270, 1998.
- [5] Ivo Senjanović. Building of ships and floating structures on horizontal berth and their launching by tipping table principle. In *The Proceedings of the International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 1999.
- [6] Ivo Senjanović et al. Side launching of ships from horizontal berth by tipping table arrangement. *Brodogradnja*, 47(2):135–143, 1999.
- [7] Henry G Weller, Gavin Tabor, Hrvoje Jasak, and Christer Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, 1998.

- [8] Hrvoje Jasak, Aleksandar Jemcov, Željko Tuković, et al. OpenFOAM: A C++ library for complex physics simulations. In *International Workshop on Coupled Methods in Numerical Dynamics*, Volume 1000, pages 1–20. IUC Dubrovnik, Croatia, 2007.
- [9] Hrvoje Jasak. CFD analysis in Subsea and Marine technology. In *IOP Conference Series: Materials Science and Engineering*, volume 276, page 012009. IOP Publishing, 2017.
- [10] Pointwise. Why You Should Be Using Overset Grids. <http://www.pointwise.com/theconnector/2014-September/Why-You-Should-Be-Using-Overset-Grids.html>.
- [11] Celeritas Simulation Technology, LLC What is Overset? http://celeritassimtech.com/?page_id=15
- [12] Hidajet Hadžić. *Development and application of finite volume method for the computation of flows around moving bodies on unstructured, overlapping grids*. PhD thesis, Technische Universität Hamburg, 2006.
- [13] R. Noack. Suggar ++ Capabilities and Introduction on Usage. http://2014.oversetgridsymposium.org/assets/presentations/tutorials/OGS2014_Suggar_Tutorial.pdf.
- [14] STAR-CCM+. Star-ccm+ overset mesh. <https://mdx2.plm.automation.siemens.com/sites/default/files/flier/pdf/Siemens-PLM-CD-adapco-STAR-CCM-overset-mesh-fs-59886-A2.pdf>
- [15] NASA. Pegasus 5 Documentation. <https://www.nas.nasa.gov/publications/software/docs/pegasus5/index.html>.
- [16] SHARCNET. Overset Domain Connectivity. https://www.sharcnet.ca/Software/Ansys/17.0/en-us/help/flu_ug/flu_ug_sec_overset_hole_cutting.html.
- [17] Tomislav Marić, Holger Marschall, and Dieter Bothe. voFoam - A geometrical Volume of Fluid algorithm on arbitrary unstructured meshes with local dynamic adaptive mesh refinement using OpenFOAM *arXiv preprint arXiv:1305.3417*, 2013.

- [18] H. Jasak and V. Vukčević. Native Overset Mesh Library for OpenFOAM. Slides from Overset Training (OFW12).
- [19] V. Škurić. Release of Overset Mesh in foam-extend. <https://foam-extend.fsb.hr/blog/2017/01/17/release-of-overset-mesh-in-foam-extend/>.
- [20] Wikipedia. Inverse distance weighting. https://en.wikipedia.org/wiki/Inverse_distance_weighting.
- [21] Pablo M Carrica, Robert V Wilson, Ralph W Noack, and Fred Stern. Ship motions using single-phase level set with dynamic overset grids. *Computers & Fluids*, 36(9):1415–1433, 2007.