

# Implementacija numeričkog sprezanja interakcije fluida i elastične konstrukcije

---

**Finzir, Ivica**

**Master's thesis / Diplomski rad**

**2019**

*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:235:961942>

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

*Download date / Datum preuzimanja:* **2024-11-24**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

## DIPLOMSKI RAD

Ivica Finzir

ZAGREB, 2019.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

## DIPLOMSKI RAD

IMPLEMENTACIJA NUMERIČKOG SPREZANJA INTERAKCIJE FLUIDA I  
ELASTIČNE KONSTRUKCIJE

Mentor:

Doc.dr.sc. Dubravko Matijašević

Student:

Ivica Finzir

ZAGREB, 2019.

*Zahvaljujem se svojoj obitelji i djevojci  
Andriani na podršci tijekom studiranja.  
Također zahvaljujem se mentoru doc.dr.sc  
Dubravku Matijaševiću na pomoći i stručnim  
savjetima tijekom izrade ovog rada.*

## **Izjava**

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

Zagreb, srpanj 2019.

Ivica Finzir



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE  
Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne i diplomske ispite studija zrakoplovstva



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum: -07- 2019	Prilog
Klasa: 002-C4/19-G/3	
Ur.broj: 15-1703-19-283	

## DIPLOMSKI ZADATAK

Student: Ivica Finzir

Mat. br.: 0035191509

Naslov rada na hrvatskom jeziku: **Implementacija numeričkog spreznja interakcije fluida i elastične konstrukcije**

Naslov rada na engleskom jeziku: **A Numerical Fluid-Structure Interaction Coupling Implementation**

Opis zadatka:

Promatra se problem numeričkoga opisa interakcije fluida i elastične konstrukcije ukliještene u jednom dijelu ruba, pri kojemu materijal konstrukcije ima barem 3 reda veličine veću gustoću mase od fluida. Za numerički model domene fluida potrebno je primijeniti OpenFoam, a za elastičnu konstrukciju Calculix programske pakete. U okviru ovog rada potrebno je napisati algoritam za spregu navedena dva programa.

U radu je potrebno:

- Proučiti specifičnosti numeričkog spreznja domena fluida i elastične konstrukcije za promatrani problem.
- Napisati algoritam za prenošenje opterećenja, s mreže kontrolnih volumena na mrežu konačnih elemenata.
- Napisati algoritam za prenošenje pomaka, s mreže konačnih elemenata na mrežu kontrolnih volumena.
- Validirati algoritam na nekom od uobičajenih primjera iz literature, interpretirati dobivene rezultate.
- Primijeniti algoritam na iznalaženje numeričkog rješenja strujanja oko elastičnog krila, ili nekog drugog tipičnog zrakoplovnog problema kojemu je metoda primjerena.

U okviru rada potrebno je prezentirati problem i korištene metode, komentirati dobivene rezultate te navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

2. svibnja 2019.

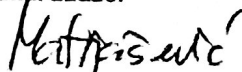
Rok predaje rada:

4. srpnja 2019.

Predviđeni datumi obrane:

10., 11. i 12. srpnja 2019.

Zadatak zadao:

  
Doc. dr.sc. Dubravko Matijašević

Predsjednik Povjerenstva:

  
Prof. dr. sc. Milan Vrdoljak

# Sadržaj

Sadržaj	v
Popis slika	vii
Popis tablica	ix
Popis oznaka	x
Sažetak	xii
Summary	xiii
<b>1. Uvod</b>	<b>1</b>
<b>2. Metoda konačnih elemenata - CalculiX</b>	<b>3</b>
<b>3. Metoda kontrolnih volumena - OpenFoam</b>	<b>5</b>
3.1. ALE formulacija . . . . .	5
3.2. Metoda kontrolnih volumena . . . . .	5
3.2.1. Vremenska diskretizacija . . . . .	9
3.2.2. Rubni uvjeti . . . . .	9
3.2.3. Diskretizacija Navier-Stokesovih jednažbi . . . . .	10
<b>4. Interakcija fluida i elastične konstrukcije</b>	<b>11</b>
4.1. Sučelje (engl. Interface) . . . . .	13

4.1.1. Konvencionalni serijski algoritam . . . . .	14
4.1.2. Implicitni algoritmi . . . . .	15
4.2. Komunikacija . . . . .	17
4.3. Interpolacija s mreže na mrežu . . . . .	20
4.3.1. Interpolacija radijalnim baznim funkcijama . . . . .	20
<b>5. Rezultati</b>	<b>26</b>
5.1. Dvodimenzionalna prepreka - 1. slučaj . . . . .	26
5.2. Dvodimenzionalna prepreka - 2. slučaj . . . . .	31
5.3. Trodimenzionalni slučaj s vertikalnim stupom . . . . .	35
5.4. Polukrilo . . . . .	39
<b>6. Zaključak</b>	<b>43</b>
<b>Literatura</b>	<b>44</b>



# Popis slika

4.1	Podjela interakcije fluida i elastične konstrukcije . . . . .	12
4.2	Konvencionalni serijski algoritam . . . . .	14
4.3	Server klijent komunikacija . . . . .	18
4.4	Uspostava veze između servera i klijenta . . . . .	19
5.1	Mreža kontrolnih volumena prepreke - 1. slučaj . . . . .	27
5.2	Mreža konačnih elemenata 1. slučaja . . . . .	28
5.3	Usporedba pomaka vrha za PRECICE i implementirani algoritam . . . . .	28
5.4	Polje brzina u vremenskom trenutku 0.4 s . . . . .	29
5.5	Polje brzina u vremenskom trenutku 5 s . . . . .	29
5.6	Polje brzina u vremenskom trenutku 30 s . . . . .	30
5.7	Mreža kontrolnih volumena prepreke - 2. slučaj . . . . .	32
5.8	Mreža konačnih elemenata prepreke - 2. slučaj . . . . .	32
5.9	Usporedba pomaka vrha za PRECICE i implementiranog algoritam . . . . .	33
5.10	Polje brzina u vremenskom trenutku 0.1 s . . . . .	33
5.11	Polje brzina u vremenskom trenutku 0.5 s . . . . .	34
5.12	Polje brzina u vremenskom trenutku 2 s . . . . .	34
5.13	Mreža kontrolnih volumena za trodimenzionalni slučaj s vertikalnim stupom . . . . .	36
5.14	Mreža konačnih elemenata za trodimenzionalni slučaj s vertikalnim stupom . . . . .	36
5.15	Rezultati pomaka vrha za trodimenzionalni slučaj s vertikalnim stupom . . . . .	37
5.16	Polje brzina u vremenskom trenutku 2.2 s - presjek . . . . .	37
5.17	Mreža kontrolnih volumena za polukrilo - isječak . . . . .	40

5.18 Mreža konačnih elemenata za polukrilo . . . . .	41
5.19 Pomaci vrha polukrila u smjeru okomitom na strujanje . . . . .	41

# Popis tablica

4.1	Radijalne bazne funkcije . . . . .	21
5.1	Postavke simulacije za dvodimenzionalnu prepreku - 1. slučaj . . . . .	26
5.2	Rubni uvjeti na mreži kontrolnih volumena . . . . .	27
5.3	Postavke simulacije za dvodimenzionalnu prepreku - 2. slučaj . . . . .	31
5.4	Dimenzije domene kontrolnih volumena i konačnih elemenata . . . . .	31
5.5	Postavke simulacije trodimenzionalnog slučaj s vertikalnim stupom . . . . .	35
5.6	Rubni uvjeti na mreži kontrolnih volumena . . . . .	35
5.7	Dimenzije polukrila i konstrukcijskih elemenata . . . . .	39
5.8	Postavke simulacije za slučaj polukrilo . . . . .	39
5.9	Rubni uvjeti na mreži kontrolnih volumena za slučaj polukrila . . . . .	40

# Popis oznaka

$\alpha_j$	Težinski koeficijent [-] . . . . .	20
$\alpha$	Koeficijent alfa kod RBFi [-] . . . . .	20
$\beta$	Koeficijent beta kod RBFi [-] . . . . .	20
$\dot{m}_f$	Apsolutni maseni tok fluida kroz stranicu ćelije [kg/s] . . . . .	6
$\dot{V}_f$	Volumni tok stranice ćelije [m <sup>3</sup> /s] . . . . .	6
$\Gamma_\phi$	Koeficijent difuzije za svojstvo $\phi$ [-] . . . . .	8
$\ddot{\mathbf{d}}$	Akceleracija konstrukcije [m/s <sup>2</sup> ] . . . . .	3
$\dot{\mathbf{d}}$	Brzina konstrukcije [m/s] . . . . .	3
$\sigma_F$	Tenzor naprezanja na strani fluida [N/m <sup>2</sup> ] . . . . .	13
$\sigma_S$	Tenzor naprezanja na strani konstrukcije [N/m <sup>2</sup> ] . . . . .	13
$\mathbf{C}$	Matrica krutosti [N/m] . . . . .	3
$\mathbf{d}$	Pomaci konstrukcije [m] . . . . .	3
$\mathbf{d}_F$	Pomaci diskretizirane domene fluida na sučelju [m] . . . . .	13
$\mathbf{d}_f$	Vektor između P i N [m] . . . . .	8
$\mathbf{d}_S$	Pomaci diskretizirane domene konstrukcije na sučelju [m] . . . . .	13
$\mathbf{F}$	Sile površinskog opterećenja [N] . . . . .	3
$\mathbf{H}_{FS}$	Matrica transformacije pomaka [-] . . . . .	20
$\mathbf{K}$	Matrica prigušenja [Ns/m] . . . . .	3
$\mathbf{k}_f$	Neortogonalni dio jedinične normale na stranicu ćelije [ ] . . . . .	8
$\mathbf{M}$	Matrica masa [kg] . . . . .	3
$\mathbf{n}$	Jedinični vektor vanjske normale [-] . . . . .	6
$\mathbf{r}$	Vektor položaja [m] . . . . .	6

$\mathbf{v}$	Brzina gibanja kontinuuma [m/s] . . . . .	6
$\mathbf{v}_F$	Brzina domene fluida na sučelju [m/s] . . . . .	13
$\mathbf{v}_S$	Brzina domene konstrukcije na sučelju [m/s] . . . . .	13
$\mathbf{v}_s$	Brzina točaka proizvoljne površine S [m/s] . . . . .	6
$\mu$	Dinamička viskoznost fluida [Pa s] . . . . .	6
$\phi$	Opće fizikalno svojstvo [-] . . . . .	6
$\phi_f$	Vrijednost fizikalnog svojstva na stranici ćelije [-] . . . . .	6
$\rho$	Gustoća [kg/m <sup>3</sup> ] . . . . .	6
$E_S$	Youngov modul [MPa] . . . . .	42
$f_x$	Interpolacijski faktor [-] . . . . .	7
$p$	Tlak [Pa] . . . . .	13
$S_f$	Površina stranice ćelije [m <sup>2</sup> ] . . . . .	6
$t$	Vrijeme [s] . . . . .	3
$U_{inf}$	Brzina strujanja fluida [m/s] . . . . .	42

# Sažetak

U ovom radu implementiran je eksplicitni algoritam za analizu interakcije fluida i elastične konstrukcije. Metode koje su korištene su metoda konačnih elemenata za domenu konstrukcije i metoda kontrolnih volumena za domenu fluida. Korišteni su programi otvorenog koda CalculiX i OpenFoam. Za implementaciju algoritma, povezivanje domene fluida i konstrukcije, interpolaciju podataka između tih domena te međuprocesnu komunikaciju korišten je programski jezik C++. Komunikacija je ostvarena kao veza servera i klijenta.

**Ključne riječi:** interakcija fluida i elastične konstrukcije, eksplicitni algoritam, interpolacija radijalnim baznim funkcijama, međuprocesna komunikacija

# Summary

In this thesis, an explicit algorithm for fluid-structure interaction analysis was implemented. Finite element method is used for describing structure domain, while Finite volume method is used for fluid domain. The programs that were used, for each domain, are CalculiX and OpenFoam. C++ language is used for the algorithm implementation, coupling fluid and structure domains, for interpolation and interprocess communication. The communication is established in server-client paradigm.

**Keywords:** fluid-structure interaction, explicit algorithm, radial basis function interpolation, interprocess communication

# 1 | Uvod

Računalne simulacije su danas tipično relativno napredne u okviru simulacija stvarnih inženjerskih problema. Postalo je dostupno analizirati inženjerske probleme samim time što je danas dostupan vrlo velik broj alata i razvijenih metoda.

Jedan od glavnih problema kojim se inženjeri danas susreću je problem interakcije fluida s elastičnom konstrukcijom. U većini slučajeva moguće je analizirati zasebno fluid te zasebno konstrukciju no zbog sve većih zahtjeva za što realnijom numeričkom analizom promatraju se zajedno što se posebno odnosi na područje zrakoplovstva.

Ovisno o fazi razvoja zrakoplova moguće je provesti numeričku analizu različite vjernosti originalu (engl. Fidelity). U samom početku numerička analiza koja se provodi najčešće je statička analiza konstrukcije dok kod analize dinamike fluida na početku se pretpostavlja beskonačna krutost konstrukcije koja se nalazi u struji fluida pa konstrukcija ne utječe na polje tlaka i brzina. Kako se zbog različite vjernosti originalu u kasnijim fazama razvoja zahtjeva veća sličnost s stvarnim slučajem fluid i konstrukcija se više ne mogu zasebno analizirati, te je potrebno u numeričku analizu uvesti interakciju fluida i elastične konstrukcije. Takav pristup iziskuje istraživanje novih metoda numeričke analize fluida i konstrukcije. Od posebnog interesa za to područje je interpolacija radialnim baznim funkcijama i programi s otvorenim kodom zbog mogućnosti prilagodbe problemu i visoke cijene dostupnih komercijalnih programa. Za zrakoplovstvo tipični problemi koji se javljaju su problemi aeroelastičnosti koji se mogu numerički analizirati eksplicitnim algoritmima, jer je tipično da je interakcija između fluida i elastične konstrukcije slaba.

Ostatak ovog rada organiziran je na sljedeći način. U drugom poglavlju prikazane su



osnovne jednađbe koje koristi program CalculiX [1] te je u trećem poglavlju dan kratki pregled metode kontrolnih volumena za pomičnu mrežu te diskretizacijskih metoda koje će se koristiti u ovome radu i koji je dostupan u programu OpenFoam [2]. U četvrtom poglavlju je detaljno analizaran problem sučelja, dostupni eksplicitni algoritmi, interpolacija informacija s mreže na mrežu te naposljetku metode komunikacije između dva programa. U petom poglavlju analizirana su 4 slučaja koja su relativno pojednostavljena kako bi se skratilo trajanje simulacije. Prva dva slučaja prepreke u struji fluida su uspoređena s dostupnim programom otvorenog koda. U zadnjem poglavlju dan je zaključak te preporuke za nastavak rada.

## 2 Metoda konačnih elemenata - CalculiX

U ovome radu analiziran je i implementiran algoritam za slabu interakciju između fluida i konstrukcije te je stoga realno za očekivati kako će pomaci konstrukcije biti relativno mali odnosno zadržati se u geometrijski i materijalno linearnom području. U nastavku će biti ukratko opisana metoda konačnih elemenata koja je iskorištena u radu te dostupna u programu CalculiX [3]. Izrazi su svi izvedeni za Lagrangeov opis gibanja koji je tipičan za ovakve slučajeve metode konačnih elemenata.

U svom osnovnom obliku izraz metode konačnih elemenata je

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{K}\dot{\mathbf{d}} + \mathbf{C}\mathbf{d} = \mathbf{F}(t) \quad (2.1)$$

gdje je  $\mathbf{M}$  matrica masa,  $\mathbf{K}$  je matrica prigušenja te  $\mathbf{C}$  matrica krutosti. Sa  $\mathbf{F}$  označene su sile koje će u ovome radu upravo biti sile površinskog opterećenja uslijed djelovanja fluida na konstrukciju.

Metoda kojom će se izraz (2.1) riješiti u CalculiX-u spada u direktne metode. To znači da će se integracija izraza (2.1) provoditi bez promjena tog izraza. Integracija se provodi u vremenskim koracima i u svakom vremenskom koraku pokušava se postići ravnoteža sila. U direktne metode spada  $\alpha$ -metoda koja se koristi u CalculiX-u za implicitnu analizu dinamike konstrukcije što će biti objašnjeno u nastavku no najprije će biti prikazana Newmark-ova metoda jer su pretpostavke za pomake i brzinu iste kao kod  $\alpha$ -metode. Pretpostavke za pomak i brzinu [3].

$$\mathbf{d}_{i+1} = \mathbf{d}_i + (\Delta t)\dot{\mathbf{d}}_i + (\Delta t)^2\left[\left(\frac{1}{2} - \beta\right)\ddot{\mathbf{d}}_i + \beta\ddot{\mathbf{d}}_{i+1}\right] \quad (2.2)$$

$$\dot{\mathbf{d}}_{i+1} = \dot{\mathbf{d}}_i + (\Delta t)\left[(1 - \gamma)\ddot{\mathbf{d}}_i + \gamma\ddot{\mathbf{d}}_{i+1}\right] \quad (2.3)$$

Pomoću izraza (2.2) i (2.3) računa se u vremenskom trenutku  $t_{n+1}$  izraz 1.1 te se dobiva sljedeći oblik.

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1} + \mathbf{K}\dot{\mathbf{d}}_{n+1} + \mathbf{C}\mathbf{d}_{n+1} = \mathbf{F}_{n+1} \quad (2.4)$$

Budući da je sada jedina nepoznata veličina akceleracija  $\ddot{\mathbf{d}}$  može se riješiti sustav linearnih jednadžbi (2.4). Ova metoda je Newmark-ova metoda.

Problem kod Newmark-ove metode je što metoda nema numeričko prigušenje pa može postati nestabilna te metoda nemože postići drugi red točnosti pod nekim određenim uvjetima stoga je kako bi se zaobišli problemi Newmark-ove metode napravljena  $\alpha$ -metoda:

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1} + (1 + \alpha)\mathbf{K}\dot{\mathbf{d}}_{n+1} - \alpha\mathbf{K}\dot{\mathbf{d}}_n + (1 + \alpha)\mathbf{C}\mathbf{d}_{n+1} - \alpha\mathbf{C}\mathbf{d}_n = \mathbf{F}(t_n + (1 + \alpha)\Delta t). \quad (2.5)$$

Rješenjem prethodnog sustava linearnih jednadžbi dolazi se do nepoznate veličine akceleracije  $\ddot{\mathbf{d}}$ , a do pomaka i brzine se dolazi iz jednadžbi (2.2) i (2.3).  $\alpha$ -metoda je stabilna i drugog reda točnosti ukoliko je koeficijent  $\alpha$  između  $\frac{-1}{3}$  i 0 te je

$$\begin{aligned} \gamma &= \frac{1 - 2\alpha}{2} \\ \beta &= \frac{(1 - \alpha)^2}{4}. \end{aligned} \quad (2.6)$$

Ova metoda je dostupna kao što je već navedeno u programu CalculiX te je pogodna upravo zbog stabilnosti za interakciju konstrukcije i fluida te je drugog reda točnosti. U ovome radu program CalculiX će se pozivati prema potrebi od strane klijenta, odnosno klijent će poslati zahtjev za pokretanjem proračuna i osigurati ulazne podatke za CalculiX. CalculiX će izvršiti proračun te će dobivene rezultate klijent učitati u memoriju i nastaviti dalje izvršavati ostatak koda.

# 3 Metoda kontrolnih volumena - OpenFoam

## 3.1. ALE formulacija

Najčešći način formulacije metode kontrolnih volumena je u Euler-ovoj formi. Kod te forme mreža kontrolnih volumena je fiksirana u prostoru dok se kontinuum (materijalna točka) giba kroz mrežu i u pojedinim točkama promatraju se određena svojstva. Kod takve formulacije moguće su veće deformacije kontinuuma. Alternativna forma je Lagrange-ova forma kod koje se materijalne točke poklapaju i gibaju s čvorovima na mreži [4], te je puno veća preciznost praćenja materijalne točke nego kod Euler-ove metode.

Treća metoda je ALE metoda (engl. Arbitrary Lagrangian-Eulerian). Kod ALE metode mreža se može proizvoljno gibati u odnosu na kontinuum. U ovome radu za problem rješavanja strujanja koristiti će se program OpenFoam i potprogram pimpleDyMFoam odnosno diskretizacija metodom kontrolnih volumena. Budući da je jedna od osnova interakcije fluida i elastične konstrukcije deformabilna mreža kontrolnih volumena, za formulaciju metode kontrolnih volumena koristi se ALE metoda. Metoda je dostupna unutar potprograma pimpleDyMFoam.

## 3.2. Metoda kontrolnih volumena

Problem strujanja nestlačivog fluida oko elastične konstrukcije opisuje se u ALE formi [5] te je opisan Navier-Stokesovim jednadžbama koje su u integralnom obliku:

$$\oint_S \mathbf{n} \cdot \mathbf{v} dS = 0 \quad (3.1)$$

$$\frac{d}{dt} \int_V \rho \mathbf{v} dV + \oint_S \mathbf{n} \cdot \rho(\mathbf{v} - \mathbf{v}_s) \mathbf{v} dS = \oint_S \mathbf{n} \cdot (\mu \nabla \mathbf{v}) dS + \int_V \rho \mathbf{g} dV - \int_V \nabla p dV \quad (3.2)$$

Kako bi se prethodne jednadžbe mogle numerički riješiti potrebno je provesti vremensku i prostornu diskretizaciju istih. Diskretizacijom jednadžbi dobiva se sustav linearnih jednadžbi koji zapravo opisuje ono što je predstavljeno izrazima (3.1) i (3.2).

U nastavku će biti prikazana diskretizacija transportne jednadžbe za skalarno svojstvo  $\phi$  na kontrolnom volumenu  $V_P$ . Integralni oblik skalarne transportne jednadžbe je

$$\frac{d}{dt} \int_V \rho \phi dV + \oint_S \mathbf{n} \cdot \rho(\mathbf{v} - \mathbf{v}_s) \phi dS - \oint_S \mathbf{n} \cdot (\rho \Gamma_\phi \nabla \phi) dS = \int_V s_\phi dV \quad (3.3)$$

gdje je prvi član vremenski član, drugi konvekcijski, treći difuzijski te je četvrti član izvorski. Prethodna jednadžba integrira se po svakom kontrolnom volumenu  $V_P$  te prelazi u oblik

$$\frac{d}{dt} \int_{V_P} \rho \phi dV + \oint_{\partial V_P} \mathbf{n} \cdot \rho(\mathbf{v} - \mathbf{v}_s) \phi dS - \oint_{\partial V_P} \mathbf{n} \cdot (\rho \Gamma_\phi \nabla \phi) dS = \int_{V_P} s_\phi dV \quad (3.4)$$

gdje je  $\partial V_P$  granica kontrolnog volumena koji se promatra. Prostorna i vremenska raspodjela varijable  $\phi$  pretpostavlja se u obliku

$$\phi(\mathbf{r}) = \phi_P + (\mathbf{r} - \mathbf{r}_P)(\nabla \phi)_P \phi(t^o + \delta t) = \phi^o + \delta t \left( \frac{\partial \phi}{\partial t} \right)^o \quad (3.5)$$

gdje je  $\phi_P$  i  $\phi^o$  vrijednosti varijable  $\phi$  u točki P i na početku vremenskog koraka.

U osnovnoj metodi kontrolnih volumena pravilo za računanje volumnog integrala po kontrolnom volumenu je

$$\int_{V_P} \phi(\mathbf{r}) dV = \phi_P V_P \quad (3.6)$$

te pravilo za računanje zatvorenog površinskog integrala po granici ćelije P je

$$\oint_{\partial V_P} \phi(\mathbf{r}) dS = \sum_f \int_{S_f} \phi(\mathbf{r}) dS. \quad (3.7)$$

gdje je  $S_f$  površina stranice, a f stranica iz skupa koji zatvara volumen P. Pravilo za računanje integrala po stranici može zapisati kao

$$\int_{S_f} \phi(\mathbf{r}) dS = \phi_f S_f. \quad (3.8)$$

### Konvekcijski član

Sada je koristeći izraze (3.7) i (3.8) moguće provesti diskretizaciju konvekcijskog člana te ga zapisati u obliku

$$\oint_{\partial V_P} \mathbf{n} \cdot \rho(\mathbf{v} - \mathbf{v}_S)\phi dS = \sum_f \mathbf{n}_f \cdot [\rho(\mathbf{v} - \mathbf{v}_S)] S_f \phi_f \quad (3.9)$$

$$= \sum_f [\rho_f(\mathbf{n}_f \cdot \mathbf{v}_f) S_f - \rho_f(\mathbf{n}_f \cdot \mathbf{v}_s f)] \phi_f \quad (3.10)$$

$$= \sum_f (\dot{m}_f - \rho_f \dot{V}_f) \phi_f \quad (3.11)$$

gdje je  $\phi_f$  vrijednost varijable  $\phi$  na stranici  $f$ ,  $\dot{m}_f$  apsolutni maseni tok kroz stranicu te je  $\dot{V}_f$  volumni tok stranice ćelije koji treba zadovoljiti zakon održanja prostora.

Ono što je potrebno sljedeće odrediti je upravo vrijednost  $\phi_f$  na stranicama kontrolnog volumena. Vrijednost će ovisiti o vrijednostima u čvorovima ćelija na kojima je zajednička stranica  $f$ .

Postoji čitav niz razvijenih metoda kojima se određuje vrijednost  $\phi_f$ , a ovdje će biti navedene i ukratko prikazane one koje će biti korištene u radu.

Prva metoda je linearna interpolacija. To je metoda drugog reda točnosti te se može zapisati kao

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N. \quad (3.12)$$

U izrazu (3.12)  $f_x$  je interpolacijski faktor zapisan kao

$$f_x = \frac{fN}{PN}. \quad (3.13)$$

Shema koja se tipično koristi za stabilizaciju konvektivnog člana je uzvodna (engl. Upwind) shema. Ona je bezuvjetno stabilna, ali je samo 1. reda točnosti. Ova shema za razliku od linearne interpolacije uzima u obzir smjer strujanja fluida te se vrijednost na stranici uzima da je jednaka vrijednosti u centru prethodne ćelije odnosno ćelije iz čijeg smjera nailazi strujanje. Ova shema zapisuje kao

$$\phi_f = \phi_P \quad (3.14)$$

ukoliko je  $(\dot{m}_f - \rho_f \dot{V}_f) \geq 0$  te ukoliko je  $(\dot{m}_f - \rho_f \dot{V}_f) \leq 0$  kao

$$\phi_f = \phi_N. \quad (3.15)$$

Metoda je budući da uzima vrijednost susjedne ćelije prvog reda točnosti te kod upotrebe dovodi do lažne difuzije.

Kako bi se stabilizirala shema za konvekciju i istovremeno zadržala njena točnost koriste se još i TVD (engl. Total Variation Diminishing) shema. Cilj ove sheme je povećati točnost te pritom zadržati stabilnosti.

### Difuzijski član

Difuzijski član iz jednadžbe (3.2) može se diskretizirati kao

$$\oint_{\partial V} \mathbf{n} \cdot (\rho \Gamma_\phi \nabla \phi) dS = \sum_f \mathbf{n}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f S_f \quad (3.16)$$

$$= (\rho \Gamma_\phi)_f S_f \mathbf{n}_f \cdot (\nabla \phi)_f. \quad (3.17)$$

Kod prethodnog izraza  $(\nabla \phi)_f$  je gradijent na stranici  $f$ . Kod ortogonalne mreže taj se gradijent može izračunati prema izrazu

$$\mathbf{n}_f (\nabla \phi)_f = \frac{\phi_N - \phi_P}{|\mathbf{d}_f|} \quad (3.18)$$

te je u tom slučaju gradijent drugog reda točnosti.

Ukoliko mreža nije ortogonalna normalni gradijent se rastavlja na dva dijela koji se sastoje od ortogonalnog dijela i neortogonalne korekcije

$$\mathbf{n}_f (\nabla \phi)_f = \Delta_f (\nabla \phi)_f + \mathbf{k}_f (\nabla \phi)_f \quad (3.19)$$

gdje se za vektor  $\mathbf{n}_f$  može zapisati kao

$$\mathbf{n}_f = \Delta_f + \mathbf{k}_f. \quad (3.20)$$

Diskretizacija lijevog člana se provodi kao i kod ortogonalne mreže dok se desni član tretira eksplicitno o čemu se više može vidjeti u [6].

Sljedeći problem u izrazu (3.19) je gradijent  $\nabla \phi_f$  koji se može odrediti interpolacijom između gradijenata u središtu susjednim ćelijama putem centralne diferencije odnosno:

$$(\nabla \phi)_f = f_x (\nabla \phi)_P + (1 - f_x) (\nabla \phi)_N. \quad (3.21)$$

Iz jednadžbe (3.21) se vidi da je potrebno još odrediti gradijent u težištima susjednih ćelija. Taj gradijent se određuje primjenom Gaussovog integralnog teorema koji u diskretnoj formi glasi:

$$(\nabla\phi)_P = \frac{1}{V_P} \sum_f \mathbf{n}_f \phi_f S_f. \quad (3.22)$$

Još jedna opcija za određivanje gradijenta u težištima susjednih ćelija je i metoda najmanjih kvadrata prikaza u [7].

### Izvorski član

Izvorski član  $s_\phi$  se linealizira [5] kao

$$s_\phi = s_{\phi u} + s_{\phi p} \phi. \quad (3.23)$$

Sada se volumni integral po kontrolnom volumenu može izračunati kao

$$\int_{V_P} s_\phi dV = s_{\phi u} V_P + s_{\phi p} V_P \phi_P \quad (3.24)$$

### 3.2.1. Vremenska diskretizacija

Izraz (3.2) može se zapisati:

$$\frac{d}{dt}(\rho_P \phi_P V_P) = - \sum_f (\dot{m}_f - \rho_f \dot{V}_f) \phi_f + \sum_f (\rho \Gamma_\phi)_f S_f \mathbf{n}_f \cdot (\nabla\phi)_f + s_{\phi u} V_P + s_{\phi p} V_P \phi_P. \quad (3.25)$$

što predstavlja poludiskretiziranu transportnu jednadžbu. Desna strana se može zapisati kao  $F(t)$ .

Najčešća metoda vremenske diskretizacije je implicitna Eulerova metoda koja se za ovaj slučaj izražava kao

$$\frac{\rho_P^n \phi_P^n V_P^n - \rho_P^o \phi_P^o V_P^o}{\Delta t} = F(t^o) \quad (3.26)$$

koja je prvog reda točnosti i bezuvjetno stabilna.

### 3.2.2. Rubni uvjeti

Kako bi diskretizirani sustav linearnih jednadžbi koji se dobije diskretizacijom koja je prikazana u prethodnim poglavljima bio bio u potpunosti određen, potrebno je zadati



rubne uvjete. Koriste se dva rubna uvjeta, a to su Dirichletov rubni uvjet koji propisuje vrijednost varijable na rubu sustava te Neumannov koji propisuje veličinu gradijenta u smjeru normale na rubu sustava.

Dirichletov rubni uvjet na rubu domene uzima da je  $\phi = \phi_b$ . Prema [5] konvekcijski član na rubu sustava  $b$  ima vrijednost

$$(\dot{m}_b - \rho_b \dot{V}_b) \phi_b \quad (3.27)$$

gdje je  $\dot{m}_b$  apsolutni maseni tok kroz graničnu stranicu dok je  $V_b$  volumni tok granične stranice. Za difuzijski član na graničnoj stranici se može pisati

$$(\rho \Gamma_\phi)_b S_b \frac{\phi_b - \phi_P}{\mathbf{d}_n}. \quad (3.28)$$

Kod Neumanovih uvjeta konvekcijski član na graničnoj stranici ima oblik

$$(\dot{m}_b - \rho_b \dot{V}_b) (\phi_P + \mathbf{d}_n g_b) \quad (3.29)$$

dok difuzijski član ima oblik

$$(\rho \Gamma_\phi)_b S_b g_b. \quad (3.30)$$

### 3.2.3. Diskretizacija Navier-Stokesovih jednadžbi

Diskretizacija se provodi analogno postupku koji je prikazan za diskretizacija transportne jednadžbe osim konvekcijskog člana koji s najprije linealizira, a potom diskretizira. Potpuni postupak diskretizacije neće biti prikazan već će biti samo naveden diskretizirani oblik jednadžbe količine gibanja i tlaka. Diskretizirani oblik jednadžbe količine gibanja je

$$a_p \mathbf{V}_P^n + \sum_f a_N \mathbf{V}_N^n = r_P - \sum_f \mathbf{n}_f p_f^n S_f^n. \quad (3.31)$$

dok je diskretizirana jednadžba tlaka dobivena uz pomoć jednadžbe kontinuiteta Rhie-Chow interpolacijom i slijedi u nastavku.

$$\sum_f \mathbf{N}_f^n \left[ \left( \frac{1}{A_P} \right)_f (\nabla p)_f^n \right] S_f^n = \sum_f \mathbf{n}_f^n \left( \frac{\mathbf{H}_P(\mathbf{v}^n)}{A_P} \right)_f S_f^n. \quad (3.32)$$

Prethodna dva izraza rješavaju se odvojeno nekim od postojećih algoritama kao što SIMPLE algoritam, PISO algoritam te PIMPLE algoritam koji je upravo i korišten u ovome radu. Detaljniji opis SIMPLE i PISO se može pronaći u [7].

## 4 Interakcija fluida i elastične konstrukcije

Obzirom na pristup rješavanju problema interakcije fluida i elastične konstrukcije koji će se temeljiti na numeričkom pronalasku rješenja moguće je podijeliti interakciju na sljedeći način. Prvi je monolitni (engl. Monolithic) način. Kod monolitnog načina rješenja jednadžbe koje opisuju mehaniku fluida i čvrstoću sprežu se u jedinstveni sustav jednadžbi koji se nakon toga diskretizira što se može prikazati kao [8]

$$\begin{aligned}\mathbf{F}(\mathbf{v}, \mathbf{u}) &= \mathbf{0} \\ \mathbf{S}(\mathbf{v}, \mathbf{u}) &= \mathbf{0}.\end{aligned}\tag{4.1}$$

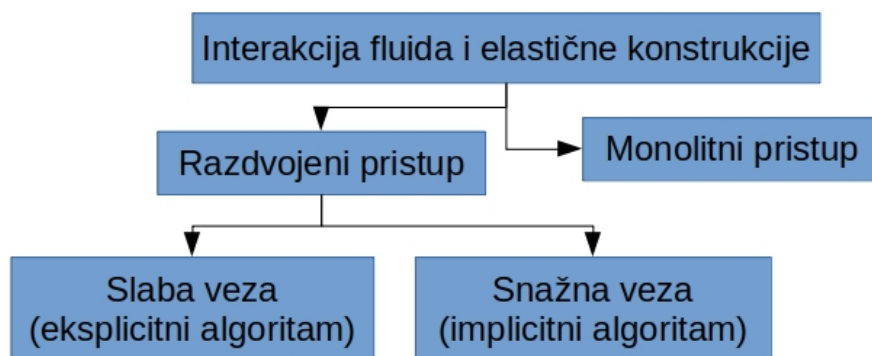
gdje  $\mathbf{F}$  značava jednadžbe dinamike fluida,  $\mathbf{S}$  označava jednadžbe dinamike konstrukcije,  $\mathbf{v}$  označava varijable dinamike fluida te  $\mathbf{u}$  označava varijable dinamike konstrukcija. Iz prethodnih jednadžbi jasno je da je rješavanje potrebno istovremeno rješavati obadvije jer svaka ovisi o varijablama dinamike fluida i dinamike konstrukcije. Problem kod ovog načina je što je za svaki novi slučaj potrebno iznova mijenjati jednadžbe koje se diskretiziraju te je takav pristup nefleksibilan i složen. Problem predstavlja i svaki slučaj sam za sebe jer je ponekad potrebno provesti modeliranje koje će uvesti niz pojednostavljenja te time slučaj udaljiti od stvarnog slučaja [9]. S druge strane ukoliko se zaobiđu prethodna ograničenja ovaj pristup osigurava konzervativnost odnosno da sustav neće umjetno povećavati ili smanjivati energiju te daje vrlo točna rješenja. Još jedan problem koji se javlja je i to što je monolitni pristup računalno zahtjevan.

Drugi način je razdvojena interakcija (engl. Partitioned). Kod ovog pristupa se svaki solver promatra odvojeno odnosno svaki kao zasebni podsustav, a sučelje ih spaja prostorno i vremenski. Na taj način je moguće iskoristiti postojeće programe i algoritme za rješavanje problema dinamike fluida i dinamike, te svaki od tih programa posebno

podešavati. Također diskretizacija se provodi potpuno odvojeno i neovisno jedno o drugome. Kod ovakvog slučaja fleksibilnost je veća no uvode se nestabilnosti o kojima će kasnije biti više rečeno. Razdvojeni način ima prednost jer se potprogrami mogu promatrati i koristiti kao black-box solveri što znači da se u njihov izvorni kod ne unose nikakve promjene već se daje informacija na ulazu i očekuje se informacija na izlazu. Jedino što je u tom slučaju još potrebno je interpolacija podataka s jedne mreže na drugu i obrnuto.

Sama razdvojena interakcija može se podijeliti u dvije skupine a to su interakcije s slabom vezom između konstrukcije i fluida, koje se rješavaju eksplicitnim algoritmima, te interakcija fluida i konstrukcije sa snažnom vezom između konstrukcije i fluida koje se rješavaju implicitnim algoritmima. Kod slabe veze konstrukcija ima nekoliko reda veličine veću gustoću od fluida dok je kod snažne veze gustoća konstrukcije približna gustoći fluida te je u nekim slučajevima i manja [10].

Ukoliko se promatra interakcija sa slabom vezom postoje slučajevi u kojima je interakcija veoma slaba pa nekad nije nužno da je dvosmjerna. To je moguće ukoliko su pomaci konstrukcije veoma mali pa ne utječu uvelike na polje brzine i tlaka kod fluida. U tom slučaju komunikacija je jednosmjerna odnosno s domene fluida interpoliraju se na domenu konstrukcije sile, a pomaci se ne prenose s domene konstrukcije na domenu fluida. Tako je postignuta veća brzina rada algoritma uz nepreveliko narušavanje točnosti. Sve navedeno prikazano je na slici 4.1.



Slika 4.1: Podjela interakcije fluida i elastične konstrukcije

## 4.1. Sučelje (engl. Interface)

Kod tipičnog problema interakcije fluida i konstrukcije sa  $S$  označena je domena konstrukcije dok je s  $F$  označena domena fluida. Sučelje između konstrukcije i fluida je označeno s  $\Gamma$ . Upravo na tom sučelju fluid i konstrukcija izmjenjuju informacije naprezanja ili tlaka te brzine. Na sučelju se zahtjevaju dva [8] rubna uvjeta.

$$\mathbf{v}_F = \mathbf{v}_S \quad (4.2)$$

$$\sigma_F \cdot \mathbf{n}_F = \sigma_S \cdot \mathbf{n}_S \quad (4.3)$$

Prvi rubni uvjet je kinematički uvjet koji zahtjeva jednakost brzina na sučelju. Ovaj rubni uvjet zahtjeva da brzine budu iste s obje strane sučelja. Budući da se mreže ne podudaraju jer se radi o različitim metodama i načinima pristupanja rješavanju problema, nije trivijalno prenesti informaciju o brzini s konstrukcije na domenu fluida stoga je potrebno provesti interpolaciju brzina s s domene konstrukcije na domenu fluida. Ovaj rubni uvjet je Dirichletov rubni uvjet [11].

Drugi rubni uvjet je uvjet jednakosti opterećenja na sučelju. Ovaj rubni uvjet je složeniji od prethodnog jer je potrebno prenesti opterećenja koja nastaju uslijed djelovanja fluida na konstrukciju te pritom zadovoljiti uvjete konzervativnosti. Konzervativnost se odnosi na to da se ukupni virtualni rad na konstrukciji ne razlikuje od virtualnog rada koji izvrši fluid što se može izraziti kao

$$\int_{\Gamma_F} \mathbf{d}_F \cdot (p_F \mathbf{n}_F) dS = \int_{\Gamma_S} \mathbf{d}_S \cdot (p_S \mathbf{n}_S) dS \quad (4.4)$$

Budući da je iz prethodnih izraza jasno kako će izlaz na OpenFoam-u biti sile koje se interpoliraju na čvorove mreže konačnih elemenata u CalculiX-u, a izlaz iz CalculiX-a pomaci u čvorovima mreže konačnih elemenata koji se interpoliraju na čvorove ćelija mreže konačnih volumena to se može prikazati [8] kao

$$F(\mathbf{d}) = \mathbf{f} \quad (4.5)$$

$$S(\mathbf{f}) = \mathbf{d} \quad (4.6)$$

gdje  $F$  i  $S$  apstraktno prikazuju niz operacija kojima se dolazi do rješenja problema dinamike fluida i konstrukcije. Prethodna dva izraza mogu se izraziti kao

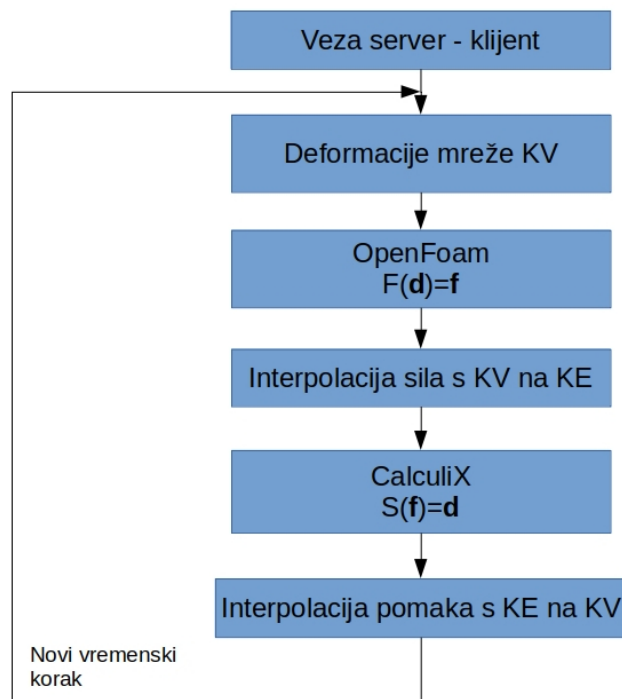
$$\mathbf{d} = S \circ F(\mathbf{d}) \quad (4.7)$$

ili

$$R(\mathbf{d}) = S \circ F(\mathbf{d}) - \mathbf{d} = 0 \quad (4.8)$$

#### 4.1.1. Konvencionalni serijski algoritam

Algoritam odabran i implementiran u ovome radu je prvog reda točnosti [8]. Implementiran je na način da se OpenFoam ponaša kao server koji započinje krug iteracije. Dobivene rezultate, odnosno sile i koordinate točaka težišta rubnih površina u kojima djeluju sile, OpenFoam kao server šalje klijentu koji obavlja interpolaciju i interpolirane sile daje na ulaz CalculiX-u. Nakon završetka rada CalculiX-a klijent pomake zajedno s točkama u kojima su poznati pomaci vraća nazad serveru. Server zaprima podatke te obavlja interpolaciju pomaka te pomiče mrežu pomoću dostupnih metoda za deformaciju mreže implementiranih u OpenFoam. Ovaj postupak i implementacija je prikazan na slici 4.2.



Slika 4.2: Konvencionalni serijski algoritam

Uz ovaj standardni algoritam postoje još drugi razvijeni algoritmi. Jedan od algoritama koji se koristi je poboljšani serijski algoritam (engl. Improved Serial Staggered) [12]. Kod tog algoritma postoji razlika u odnosu na prethodni što se vremenski koraci solvera ne podudaraju. Tako se rješenje za polje tlaka i brzina traži u vremenskim trenucima  $t^{n+\frac{1}{2}}$  dok se rješenje problema dinamike konstrukcije traži u vremenskim trenucima  $t^{n+1}$  [8].

Prethodna dva algoritma spadaju u grupu eksplicitnih algoritama što znači da nema poditeracije u svakom koraku i provjere konvergencije već se nakon svakog kruga započinje novi vremenski korak. Takvi algoritmi su primjenjivi u slučajevima kod kojih postoji veća razlika između gustoće fluida i gustoće konstrukcije što je upravo slučaj kod aeroelastičnih pojava. Ukoliko je ta razlika manja pojačava se utjecaj pojave koja se naziva efekt dodane mase [13]. Do efekta dodane mase dolazi zbog toga što konstrukcija tijekom pomaka ubrzava i dio fluida te to dovodi do nestabilnosti. Kod slučajeva gdje je gustoća konstrukcije puno veća od gustoće fluida efekt dodane mase je vrlo mali te se može potpuno zanemariti kao što je to primjer kod interakcije zrakoplovnih konstrukcija s zrakom. Prema [14] sljedeće slučajevi dovode do povećanog utjecaja dodane mase:

- Omjer gustoće konstrukcije i fluida je približno 1 ili manje
- Povećanje viskoznosti fluida
- Smanjenje krutosti konstrukcije
- Smanjenje vremenskog koraka kod implicitnih algoritama

#### 4.1.2. Implicitni algoritmi

Druga skupina algoritama su implicitni algoritmi. Ti algoritmi napravljeni su tako da postižu ravnotežu sila i pomaka unutar vremenskog koraka. Na taj način konzervativnost je puno bolje očuvana nego kod eksplicitnih algoritama te se njihova primjena lakše proširi i na područje interakcije fluida i konstrukcije gdje je gustoća konstrukcije jednaka ili manja od gustoće fluida. Ova vrsta algoritama se najviše približava po točnosti monolitnim metodama. Najčešći implicitni algoritam je Gauss-Seidel serijski iteracijski algoritam [8] u kojem se unutar svakog vremenskog koraka provodi se niz poditeracija dok se ne postigne željena konvergencija nakon čega se prelazi na idući vremenski korak.

Kod ovog algoritma specifično je to što se svaki solver pokreće tek nakon što je prethodni solver završio s radom. Vrlo sličan algoritam ovome je Jacobi paralelni algoritam kod kojeg solveri rade paralelno te tek nakon završetka jednog i drugog međusobno izmjene informacije. Međutim Gauus-Seidel iteracijski algoritam kao i paralelni Jacobi algoritam pokazuju određene probleme kod stabilnosti kao i eksplicitni algoritmi.

## 4.2. Komunikacija

Kako bi se informacije mogle izmijeniti između dva programa potrebno je uspostaviti komunikaciju između njih. Takav način komuniciranja unutar Linux operativnog sustava se naziva komunikacija između procesa (engl. Interprocess Communication) [15]. Komunikacija može biti ostvarena na više načina. U nastavku su opisani neki najčešći načini uspostave komunikacije između dva ili više procesa odnosno programa.

Prvi i najjednostavniji oblik komunikacije je komunikacija putem datoteka. Kod komunikacije putem datoteka jedan program ispisuje datoteku s informacijama nakon čega drugi program pristupa toj datoteci i čita informacije te obrnuto. Ovaj oblik komunikacije je veoma spor i nefleksibilan.

Drugi način komunikacije je putem zajedničke memorije (engl. Shared Memory). Dio memorije se alokira kao posebni dio memorije prema kojem će odabrani programi ili procesi imati pristup te biti u mogućnosti mijenjati podatke na tom dijelu memorije, a ostali koji također imaju pristup će tu promjenu vidjeti. Naknadno je još potrebno provesti sinkronizaciju pristupanja memoriji jer se ne smije dogoditi da dva programa pristupe u istom trenutku memoriji.

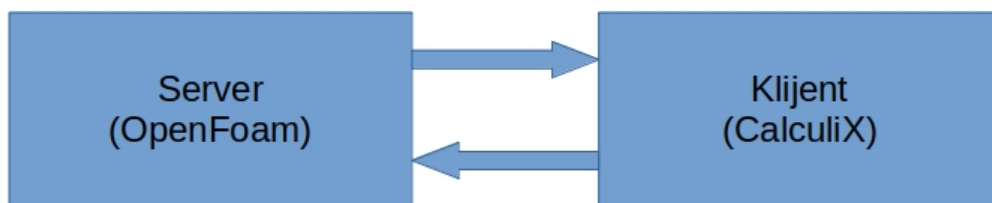
Treći i najkorišteniji način komunikacije je komunikacija putem spojne točke (engl. Socket Communication) [16]. Spojna točka se kreira te se koristi za komunikaciju između dva programa ili dva procesa. Svaki proces ili program ima svoju spojnu točku te se međusobno povezuju i izmjenjuju informacije i podatke. Kod ovog načina komunikacije je specifično to što je moguća i komunikacija između dva ili više računala što kod drugih načina komunikacije nije moguće.

Spojne točke mogu se podijeliti na Internet spojne točke te UNIX spojne točke. Karakteristično za UNIX spojne točke je to što se one nalaze na lokalnoj razini odnosno služe za komunikaciju unutar računala. Kod internet spojnih točaka moguća je komunikacija unutar računala, ali i putem mreže. Potrebno je još definirati i protokol koji će se koristiti kod internet spojnih točaka. Protokol definira na koji način će se podaci prenositi. Tipičan protokol je TCP/IP protokol.

U okviru ovoga rada odabrana je komunikacija putem spojnih točaka za izmjenu podataka između OpenFoam-a i CalculiX-a. Implementacija je provedena koristeći standardni API (Aplikacijsko programsko sučelje) za spojne točke na UNIX sustavima. Komunikacija će biti napravljena u obliku servera i klijenta. U ovome radu server će

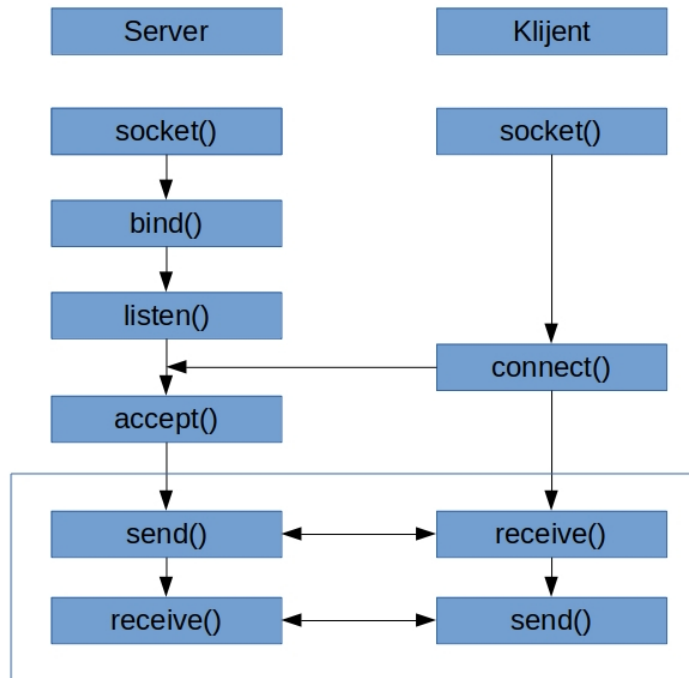


pozivati OpenFoam i interpolator pomaka, a klijent CalculiX i interpolator sila. Server je *iterativni* što znači da server čeka na odgovor klijenta te nastavlja izvršavati zadane radnje tek nakon odgovora klijenta. Na taj način je postignuta sinkronizacija između OpenFoam-a i CalculiX-a.



Slika 4.3: Server klijent komunikacija

Aplikacijsko programsko sučelje za spojne točke je programirano u programskom jeziku C, a sastoji se od niza funkcija kojima se pokreće i podešava spojna točka. Na slici 4.4 prikazan je redoslijed i način pozivanja funkcija kod kreiranja spojne točke servera i klijenta.



Slika 4.4: Uspostava veze između servera i klijenta

Funkcijom *socket()* kreira se spojna točka, dok se funkcijom *close()* zatvara. Pozivom funkcije *bind()* kreiranoj spojnoj točki se dodjeljuje adresa dok pozivom funkcije *listen()* spojna točka stavlja u poziciju čekanja. *Connect()* na strani klijenta pokušava se spojiti na spojnu točku servera. Do spajanja dolazi tek nakon izvršenja funkcije *accept()* koji kreira novu spojnu točku koja je povezana s spojnom točkom klijenta dok originalna spojna točka kod servera i dalje postoji te čeka na novi pokušaj spajanja nekog drugog klijenta na taj server. Preostale dvije funkcije su *send* i *receive* koje služe za slanje informacija između servere i klijenta. Na ovaj način te ovim redoslijedom je komunikacija implementirana u ovome radu.

Prednost korištenja ovakvog načina povezivanja je u tome što server i klijent nemoraju nužno biti pisani u istom programskom jeziku, a komunikacija između njih je i dalje moguća što povećava fleksibilnost.

### 4.3. Interpolacija s mreže na mrežu

Budući da su metoda kontrolnih volumena i metoda konačnih elemenata u načelu različite nije moguće jednostavno prenesti sile i pomake s jedne mreže na drugu. Dodatni problem javlja se kod mreža koje se međusobno ne podudaraju na zajedničkom rubu. Naime metoda kontrolnih volumena tipično za pronalaženje kvalitetnog rješenja iziskuje puno gušću mrežu nego metoda konačnih elemenata. Potrebno je stoga najprije obraditi dobivene veličine te ih nakon toga primijeniti na mrežu.

Interpolacija se može provesti na dva glavna načina [17]. To je konzistentna interpolacija te konzervativna interpolacija. Interpolacijske metode koje se najčešće koriste su metoda najbližeg susjeda i interpolacija radijalnim baznim funkcijama. Kod interpolacije važno je voditi se time da energija u sustavu bude očuvana odnosno da se energija predana konstrukciji ne povećava u odnosu na energiju koju fluid gubi.

Konzistentna interpolacija osigurava da su konstantne vrijednosti točno interpoliraju između sučelja te se koristi za interpolaciju tlaka i pomaka. Konzervativna interpolacija ima za cilj očuvati virtualni rad, no međutim to nije često moguće već će se uvijek javiti određena razlika. Kod konzervativne interpolacije ukoliko se interpolira sila s mreže na mrežu moguće su pojave u oscilaciji [10] sile prenesene na mrežu konačnih elemenata.

#### 4.3.1. Interpolacija radijalnim baznim funkcijama

Interpolacija radijalnim baznim funkcijama je vrlo moćan alat za interpoliranje složenih problema jer je implementacija takve interpolacije relativno jednostavna. Interpolant je oblika [18]

$$s(\mathbf{x}) = \sum_{j=1}^n \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + p(\mathbf{x}) \quad (4.9)$$

gdje je  $\alpha_j$  težinski koeficijent,  $p(\mathbf{x})$  polinom,  $n$  broj centara podataka,  $\phi()$  bazna funkcija te  $\|\mathbf{x} - \mathbf{x}_j\|$  Euklidska udaljenost između točaka. Težinski koeficijent određuju se iz točaka u kojima je vrijednost funkcije poznata:

$$s(\mathbf{x}_j) = g_j. \quad (4.10)$$

Ukoliko se prethodni izraz uvrsti u izraz (4.9) dobije se sustav linearnih jednadžbi u obliku

$$\begin{bmatrix} \phi & \mathbf{p} \\ \mathbf{p}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \quad (4.11)$$

Sustav jednadžbi naveden u (4.11) bit će temelj za interpolaciju u ovome radu.

Neke od radijalnih baznih funkcija koje se najčešće koriste navedene su u tablici 4.1.

Tablica 4.1: Radijalne bazne funkcije

Radikalne bazne funkcije	
Gaussova	$e^{-\epsilon r^2}$
Multikvadratna	$\sqrt{1 + (\epsilon r)^2}$
Inverzna multikvadratna	$\frac{1}{\sqrt{1 + (\epsilon r)^2}}$
Inverzna kvadratna	$\frac{1}{1 + (\epsilon r)^2}$

Radikalne bazne funkcije se mogu podijeliti u dvije skupine. Prva skupina su globalne bazne funkcije, a druga skupina su radikalne bazne funkcije na kompaktnom nosaču [19]. Kompaktne radikalne bazne funkcije dovode do slabo popunjenih matrica te mogu imati problema s gustoćom dostupnih podataka dok globalne dovode gusto popunjenih matrica koje nemaju problema s gustoćom podataka no međutim mogu postati loše uvjetovane za puno točaka. Ukoliko se radi kod globalnih funkcija o matrici manjih dimenzija problem traženja inverza se rješava vrlo brzo računalno. Problem se javlja ukoliko se radi o velikom broju poznatih točaka interpolirane funkcije što dovodi do velike matrice za koju treba pronaći inverz što postaje računalno vrlo zahtjevno i dugotrajno. Neka od rješenja za taj problem ponuđena su u radu [20]. U ovome radu to nije implementirano već se dobiveni sustav linearnih jednadžbi rješava dostupnim metodama koje su implementirane u C++ biblioteci *Eigen* [21]. U daljnim poglavljima prikazan je algoritam za prijenos sile s mreže kontrolnih volumena na mrežu konačnih elemenata te prijenos pomaka s mreže konačnih elemenata na mrežu kontrolnih volumena.

U tablici X svaka radikalna bazna funkcija ima i parametar oblika označen s  $\epsilon$ . pomoću parametra oblika kontrolira se ravnoća odnosno izbočenost oblika funkcije [22]. Vrijednosti koje parametar oblika postiže su veće od 0. Kod interakcije fluida i konstrukcije parametar oblika se koristi prema [10] najčešće između 0.001 i 0.00001 te se stoga i u ovome radu taj parametar se nalazi između navedenih vrijednosti. U prilog

tome ide i rad [22] gdje je pokazano da je greška kod interpolacije neke plohe najmanje kod manjih vrijednosti faktora oblika, a raste s porastom faktora oblika.

### Interpolacija pomaka

Za prijenos pomaka s mreže konačnih elemenata na mrežu kontrolnih volumena koristi se konzistentni način interpoliranja pomoću radijalnih baznih funkcija. U nastavku je opisan način na koji je implementirano u kodu. Za radijalnu baznu funkciju je odabrana multikvadratna funkcija sa vrijednošću parametra oblika  $\epsilon = 0.0001$ . Sam problem interpolacije može se prikazati u obliku

$$\mathbf{d}_F = \mathbf{H}_{FS} \mathbf{d}_S \quad (4.12)$$

odnosno potrebno je pronaći matricu transformacije  $\mathbf{H}_{FS}$ . Matrica transformacije se za konzistentni oblik interpolacija pronalazi vrlo jednostavno preko izraza (4.9) gdje je  $p$  linearni polinom. Izraz poprima oblik naveden u nastavku iz kojeg se dolazi do koeficijenata  $\boldsymbol{\alpha}$  i  $\boldsymbol{\beta}$ .

$$\begin{bmatrix} \boldsymbol{\phi}_{SS} & \mathbf{p} \\ \mathbf{p}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_S \\ 0 \end{bmatrix} \quad (4.13)$$

Nakon što se odrede koeficijenti interpolacija se izvršava pomoću izraza

$$\mathbf{d}_F = \begin{bmatrix} \boldsymbol{\phi}_{FS} & \mathbf{p}_F \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \quad (4.14)$$

ili ako se uvrsti (4.13) u (4.14)

$$\mathbf{d}_F = \begin{bmatrix} \boldsymbol{\phi}_{FS} & \mathbf{p}_F \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{SS} & \mathbf{p} \\ \mathbf{p}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g}_S \\ 0 \end{bmatrix}. \quad (4.15)$$

Skraćeno se prethodno može zapisati kao

$$\mathbf{d}_F = \begin{bmatrix} \mathbf{A}_{FS} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{SS} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{d}_S \end{bmatrix} \quad (4.16)$$

## Implementacija u C++

U samom C++ prethodno je implementirano u obliku klase naziva *Rbf* koja se sastoji od dvije funkcije, a to su  $X$  i  $B$ . Od klase *Rbf* kreira se objekt te se pozivaju funkcije članice klase. Prva funkcija  $X$  računa sustav linearnih jednadžbi dobiven iz izraza (4.13). Ulazni podaci u funkciju su koordinate pozicija gdje su poznati pomaci konstrukcije te iznosi tih pomaka. Izlazni podatak je izračunati koeficijenti  $\alpha$  i  $\beta$ . Za rješenje sustava linearnih jednadžbi koristi se biblioteka *Eigen*. Koordinate pozicija pomaka i pomaci dobiju se od klijenta te se sama interpolacija provodi na serveru. Druga funkcija  $B$  kreira lijevu stranu izraza (4.15) te se naposljetku množenjem dobivenog rezultata s dobivenim koeficijentima  $\alpha$  i  $\beta$  dobivaju pomaci na mreži kontrolnih volumena.

## Interpolacija sila

Kao što je navedeno već u prethodnom poglavlju za prijenos sile koristiti će se konzervativni način interpolacije zbog potrebe za očuvanjem energije na sučelju. Ovaj algoritam implementiran je iz [23] i proveden pomoću interpolacije radijalnim baznim funkcijama.

U nastavku će biti opisano kako je konzervativna interpolacija izvedena pomoću matrice transformacije slično kao što je to izvedeno za konzistentnu interpolaciju. Sile kod mreže kontrolnih volumena određene su na površini ćelija dok je kod mreže konačnih elemenata sile određene u čvorovima elemenata

$$\begin{aligned}\mathbf{F}_F &= \int_{\Gamma_F} \mathbf{t}_F d\Gamma \\ \mathbf{F}_S &= \int_{\Gamma_S} \mathbf{t}_S d\Gamma\end{aligned}\tag{4.17}$$

Kako bi se mogle odrediti prenesene sile potrebno je uvesti zahtjev za jednakošću virtualnih radova

$$\int_{\Gamma_F} \mathbf{d}_F \cdot \mathbf{t}_F d\Gamma_F = \int_{\Gamma_S} \mathbf{d}_S \cdot \mathbf{t}_S d\Gamma_S\tag{4.18}$$

gdje je  $\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}$ . Aproximacije za diskretne vrijednosti opterećenja i pomaka mogu se izraziti na sljedeći način.

$$\begin{aligned} \mathbf{d}(\mathbf{x}) &\approx \sum_{i=1}^{n^d} N_d^i(\mathbf{x})_d^i \mathbf{D}^i \\ \mathbf{t}(\mathbf{x}) &\approx \sum_{i=1}^{n^t} N_t^i(\mathbf{x})_t^i \mathbf{T}^i \end{aligned} \quad (4.19)$$

Koristeći izraze (4.19) u izrazu (4.18) dobiva se

$$\left[ M_{FF} \mathbf{D}_F \right]^T \mathbf{T}_F = \left[ M_{SS} \mathbf{D}_S \right]^T \mathbf{T}_S \quad (4.20)$$

gdje je  $M_{FF}^{ij} = \int_{\Gamma_F} N_F^i M_F^j d\Gamma$  i  $M_{SS}^{ij} = \int_{\Gamma_S} N_S^i M_S^j d\Gamma$ . Ukoliko se uzme u obzir da je matrica transformacije pomaka (4.12) koja je prikazana ranije i ukoliko se uvrsti u izraz (4.20) dobije se globalna matrica za opterećenja

$$\mathbf{T}_S = \left[ \mathbf{M}_{FF} \mathbf{H}_{FS} \mathbf{M}_{SS}^{-1} \right]^T \mathbf{T}_F \quad (4.21)$$

U izraz (4.21) se zbog prirode konačnih elemenata može uvrstiti izrazi  $\mathbf{F}_s = M_{SS}^T \mathbf{T}_S$  te  $\mathbf{F}_F = M_{FF}^T \mathbf{T}_F$  te se nakon toga dobije globalna matrica za prijenos diskretnih vrijednosti sile te je virtualni rad jednak na obje strane sučelja.

$$\mathbf{F}_S = \mathbf{H}_{FS}^T \mathbf{F}_F \quad (4.22)$$

Prethodni izraz pokazuje da ukoliko se za prijenos sila s mreže na mrežu iskoristi matrica transformacije pomaka moguće je postići konzervativni način interpolacije. Još dodatno vrijedi i da je  $\mathbf{H}_{FS}^T = \mathbf{H}_{SF}$ .

Matrica transformacije pomaka koja se koristi za prijenos sila prikazan je već u izrazu (4.12) u prethodnom poglavlju u proširenom obliku pa će ovdje biti prikazano samo u skraćenom obliku.

$$\begin{bmatrix} \boldsymbol{\phi}_{SS} & \mathbf{p}_S \\ \mathbf{p}_S^T & 0 \end{bmatrix} \mathbf{F}_S = \begin{bmatrix} \boldsymbol{\phi}_{FS} & \mathbf{p}_F \end{bmatrix}^T \mathbf{F}_F \quad (4.23)$$

Prethodni izraz se može još dodatno skratiti te zapisati kao:

$$\left[ \mathbf{A}_{SS} \right] \mathbf{F}_S = \left[ \mathbf{B}_{FS} \right]^T \mathbf{F}_F. \quad (4.24)$$

## Implementacija u C++

Implementacija je provedena slično kao implementacija pomaka u C++. Interpolacija se sastoji od klase  $Rbf_{konz}$  koja sadrži 3 funkcije članice. Kreiranjem objekta od klase nadalje se pozivaju 3 funkcije, a to su  $matricaA$ ,  $matricaB$  i  $silakonzervativna$ . Svaka od njih ima svoje ulazne varijable. Tako je ulazna varijabla u funkciju  $matricaA$  koordinate čvorova u kojima djeluju koncentrirane sile, u funkciju  $matricaB$  su ulaz koordinate čvorova mreže konstrukcije i koordinate centra težišta stranice ćelije na sučelju. Na izlaz  $matricaA$  daje rješenje lijevu stranu izraza (4.23) odnosno  $A_{SS}$  dok  $matricaB$  na izlaz daje desnu stranu izraza (4.23) odnosno  $B_{FS}$ . Kod funkcije  $silakonzervativna$  ulaz su izlazi prethodno navedenih matrica odnosno  $\mathbf{A}_{SS}$  i  $\mathbf{B}_{FS}$  te vrijednosti sila na strani fluida odnosno  $\mathbf{F}_F$ . Nakon toga funkcija  $silakonzervativna$  rješava izraz (4.23) pomoću biblioteke *Eigen* te se dobivaju interpolirane sile koje djeluju u čvorovima mreže konstrukcije odnosno  $\mathbf{F}_S$ .

Interpolacija sila se provodi kod klijenta zbog smanjenja broja uspostave komunikacije između klijenta i servera. Budući da su interpolirane sile potrebne za CalculiX, server mora poslati klijentu samo vrijednosti sila na težištu stranica ćelija i koordinate centra težišta stranica ćelije. Podaci o koordinatama točaka kod mreže konstrukcije su već dostupni klijentu. Na taj način je potrebna samo jedna komunikacija. U protivnom bi klijent morao slati podatke o točkama mreže serveru koji bi nakon toga proveo interpolaciju te vratio dobivene rezultate klijentu što znači da je komunikacija uspostavljena dva puta.



# 5 | Rezultati

U ovom poglavlju će biti prikazani rezultati za 4 različita slučaja od kojih će prva dva slučaja biti uspoređena s dostupnom bibliotekom PRECICE [24] koja povezuje dva ili više programa te uspostavlja komunikaciju, interpolaciju i algoritme za interakciju između njih. PRECICE je biblioteka otvorenog koda.

## 5.1. Dvodimenzionalna prepreka - 1. slučaj

Prvi slučaj je prepreka koja je preuzeta iz dostupnih online materijala za PRECICE biblioteku. Ovaj primjer je tipični primjer na kojem se pokazuju mogućnosti PRECICE biblioteke stoga će ta geometrija biti pogodna za testiranje implementiranog algoritma. Varijabla od interesa je pomak vrha prepreke u ovisnosti o vremenu.

U tablici 5.1 su navedene postavke simulacije.

Tablica 5.1: Postavke simulacije za dvodimenzionalnu prepreku - 1. slučaj

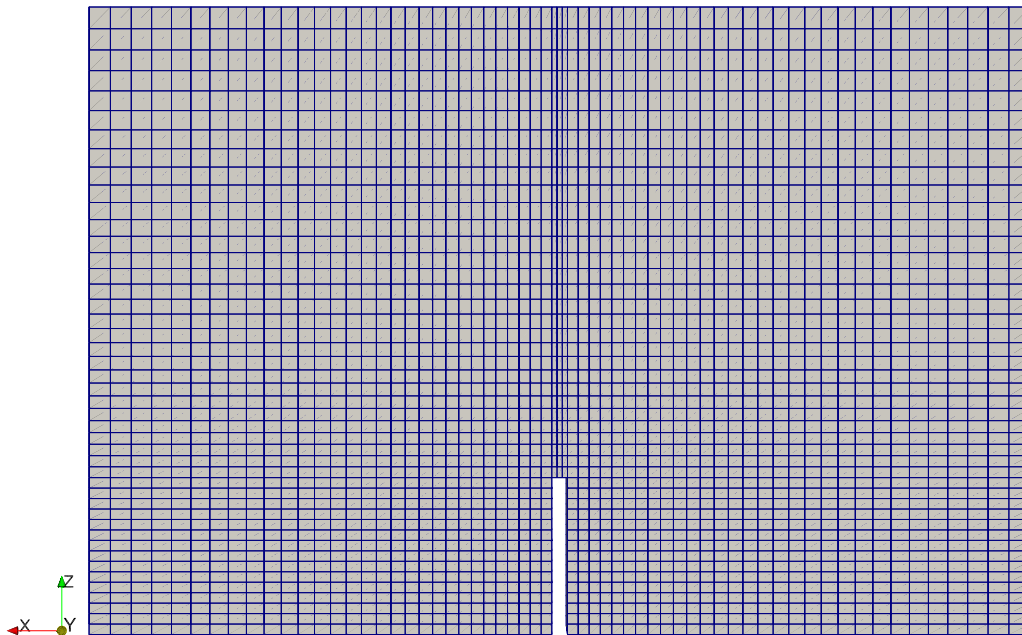
$U_{inf}$	8 m/s
$E_S$	200000 Pa
$\rho_F$	1.204 kg/m <sup>3</sup>
$\rho_S$	1000 kg/m <sup>3</sup>

Na slici 5.1 prikazana je mreža kontrolnih volumena prvog slučaja dok je na slici 5.2 prikazana mreža konačnih elemenata. Domena konačnih elemenata je diskretizirana s linearnim 3D elementima te se sastoji od 40 konačnih elemenata. Svaki element ima 2 čvora po bridu. Prepreka je uklještena na dnu te su spriječeni pomaci u y-smjeru.

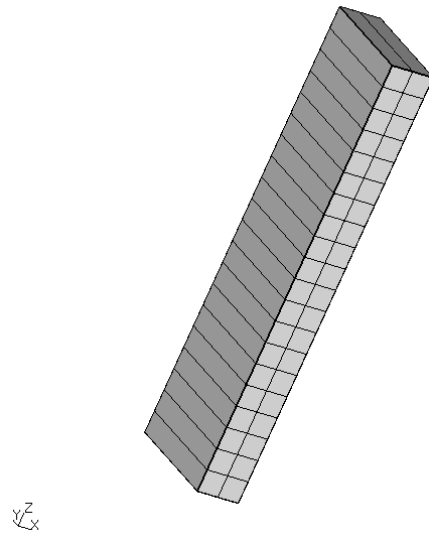
U sljedećoj tablici navedeni su rubni uvjeti i sheme diskretizacije domene fluida.

Tablica 5.2: Rubni uvjeti na mreži kontrolnih volumena

	Brzina	Tlak	Pomaci mreže
inlet	fixedValue (0 0 0)	zeroGradient	fixedValue (0 0 0)
outlet	zeroGradient	fixedValue 0	fixedValue (0 0 0)
lowerWall	fixedValue (0 0 0)	zeroGradient	fixedValue (0 0 0)
upperWall	fixedValue (0 0 0)	zeroGradient	fixedValue (0 0 0)
prepreka	movingWallVelocity	zeroGradient	fixedValue (0 0 0)
frontAndBack	empty	empty	empty

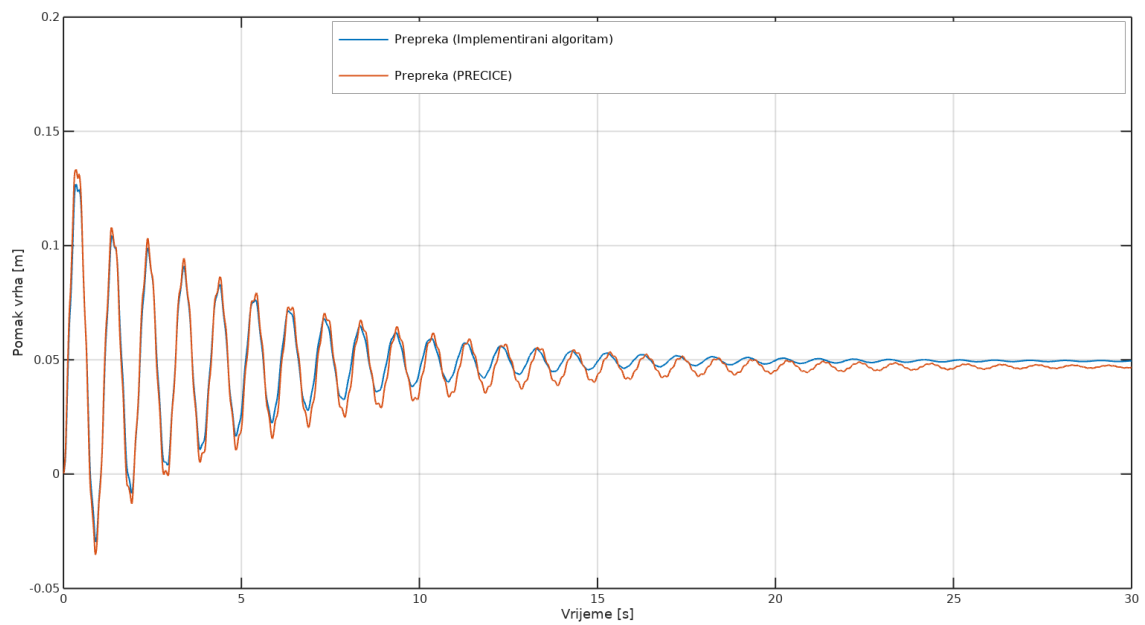


Slika 5.1: Mreža kontrolnih volumena prepreke - 1. slučaj



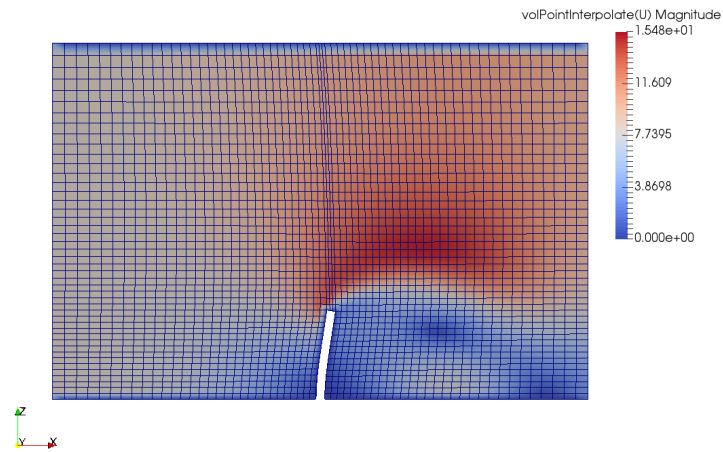
Slika 5.2: Mreža konačnih elemenata 1. slučaja

Na sljedećoj slici prikazana je usporedba pomaka vrha prepreke za PRECICE i implementiranog algoritma.

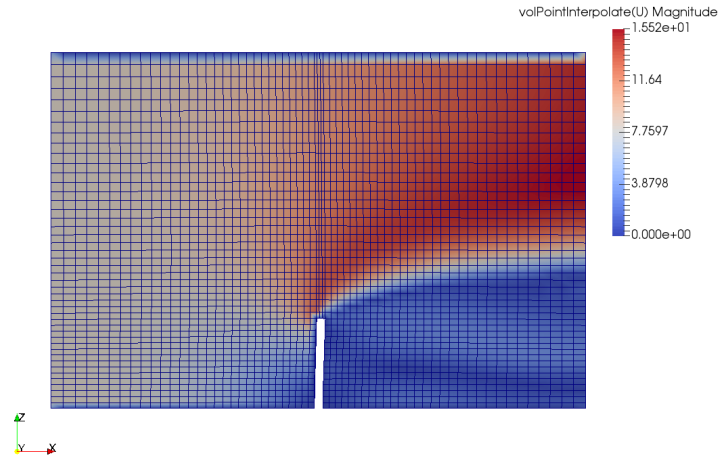


Slika 5.3: Usporedba pomaka vrha za PRECICE i implementirani algoritam

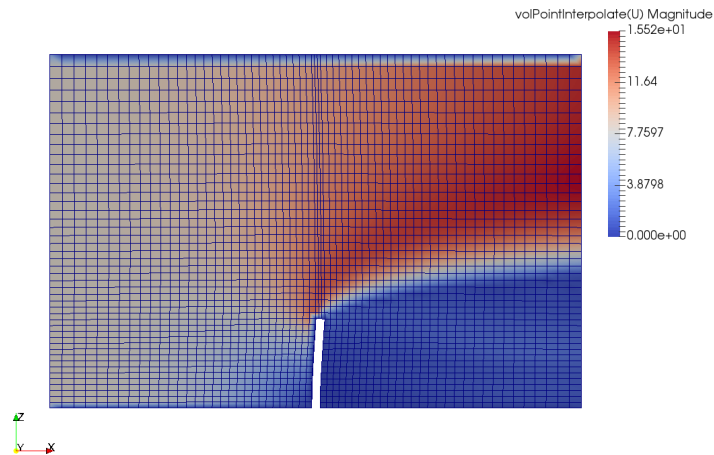
Na slikama 5.4, 5.5 i 5.6 prikazana je domena fluida s poljem brzine i deformiranom mrežom kontrolnih volumena za vremenske trenutke 0.4, 5 i 30 s.



Slika 5.4: Polje brzina u vremenskom trenutku 0.4 s



Slika 5.5: Polje brzina u vremenskom trenutku 5 s



Slika 5.6: Polje brzina u vremenskom trenutku 30 s

Ukoliko se usporede rezultati sa slike 5.3 vidi se približno poklapanje rezultata do 10-te sekunde trajanja simulacije. Nakon 10-te sekunde dolazi do manjeg odstupanje te se konačni pomak razlikuje za približno 4 milimetra. Simulacija kod implementiranog algoritma ima manje nestabilnosti te brže zauzme stacionarno stanje, dok PRECICE simulacija u 30-toj sekundi još uvijek ima blage oscilacije.

Na slici 5.4 prikazan je najveći pomak prepreke, a strujanje se još razvija. Na slici 5.5 strujanje je razvijeno no oscilacije još uvijek nisu prigušene dok je na slici 5.6 strujanje razvijeno, a prepreka je postigla konačni pomak.

## 5.2. Dvodimenzionalna prepreka - 2. slučaj

Drugi slučaj je prepreka slična prvom slučaju. Razlika je u tome što je prepreka vitkija u odnosu na prvi slučaj te je slobodni prolaz za strujanje fluida puno manji. To dovodi do veće sile na prepreku, pa da bi se zadržali u području linearnih deformacija povećanje Youngov modul. Domena konačnih elemenata je diskretizirana linearnim 3D elementima s 3 stupnja slobode po čvoru te se sastoji od 320 konačnih elemenata. Na dnu prepreke je rubni uvjet uklještenja te su spriječeni pomaci u smjeru osi z. Domena kontrolnih volumena je diskretizirana s 22150 kontrolnih volumena. Promatrati će se ponovno pomak u smjeru strujanja u ovisnosti o vremenu. Rubni uvjeti na mreži kontrolnih volumena su isti kao kod 1. slučaja prepreke. U tablici 5.3 prikazane su postavke simulacije te u tablici 5.4 dimenzije domena kontrolnih volumena i konačnih elemenata.

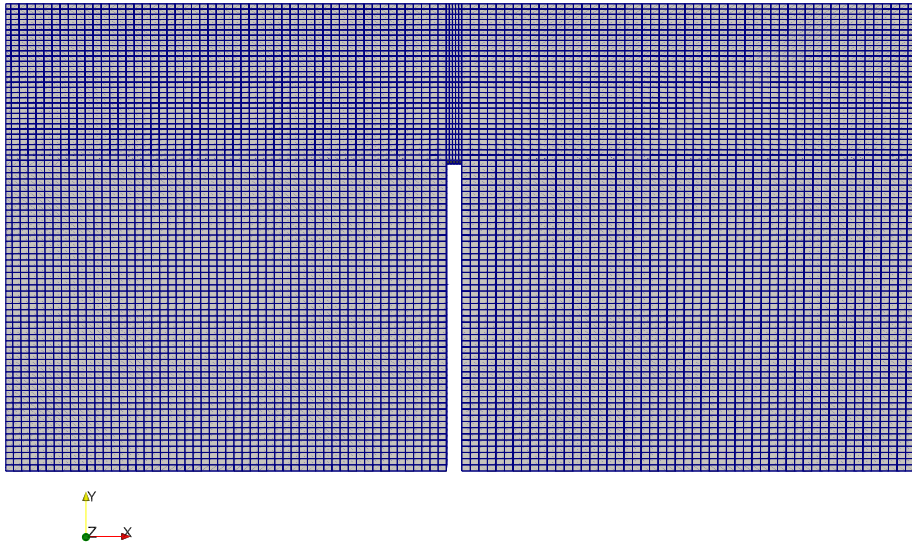
Tablica 5.3: Postavke simulacije za dvodimenzionalnu prepreku - 2. slučaj

$U_{inf}$	15 m/s
E	200 MPa
$\rho_F$	1 kg/m <sup>3</sup>
$\rho_S$	1000 kg/m <sup>3</sup>

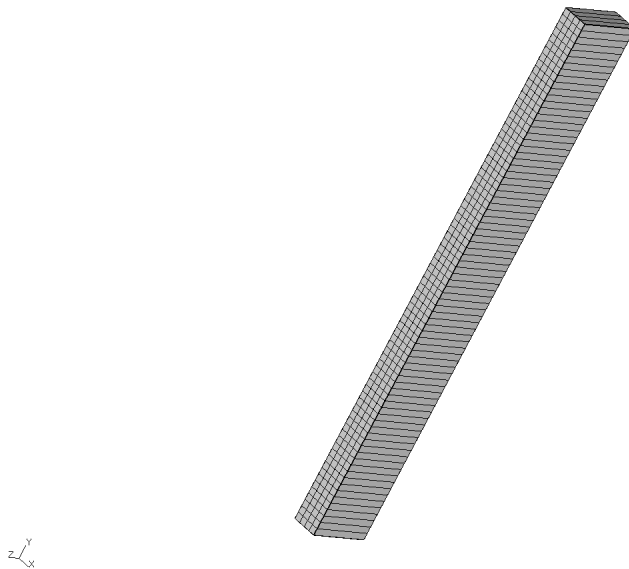
Tablica 5.4: Dimenzije domene kontrolnih volumena i konačnih elemenata

Visina domene fluida	1.2 m
Širina domene fluida	6 m
Visina prepreke	0.8 m
Širina prepreke	0.04 m

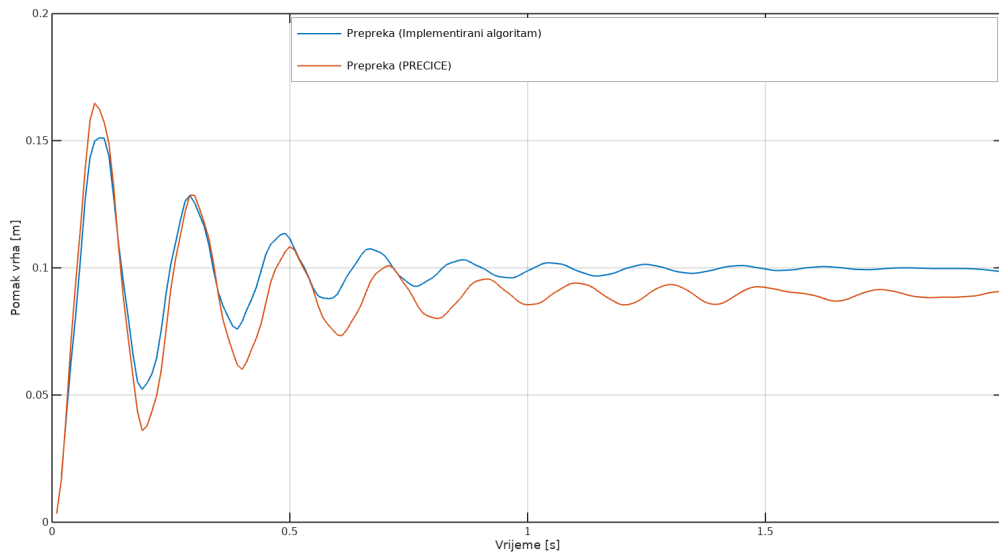
Na slikama 5.7 i 5.8 prikazane su mreža kontrolnih volumena te mreža konačnih elemenata.



Slika 5.7: Mreža kontrolnih volumena prepreke - 2. slučaj

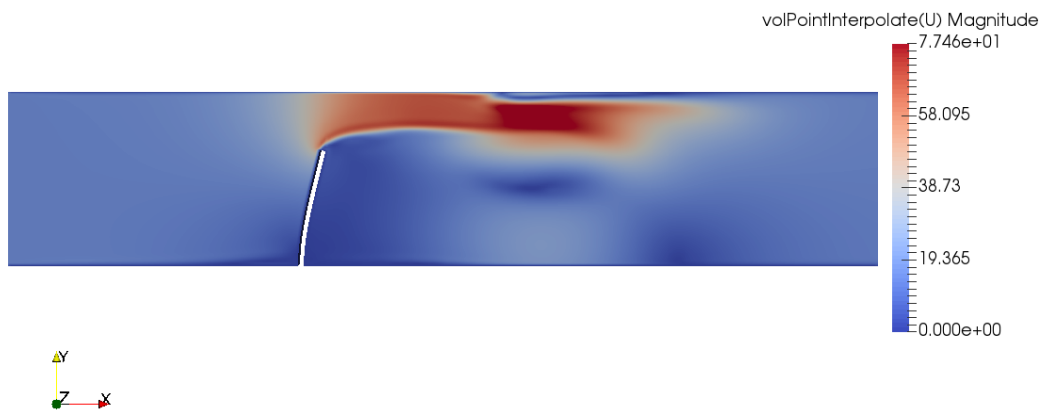


Slika 5.8: Mreža konačnih elemenata prepreke - 2. slučaj



Slika 5.9: Usporedba pomaka vrha za PRECICE i implementiranog algoritam

Konačni pomaci u ovom slučaju razlikuju se za približno 10 milimetara. Prigušenje je veće kod implementiranog algoritma koji se vrlo brzo stabilizira dok kod PRECICE oscilacije još uvijek postoje.



Slika 5.10: Polje brzina u vremenskom trenutku 0.1 s





Slika 5.11: Polje brzina u vremenskom trenutku 0.5 s



Slika 5.12: Polje brzina u vremenskom trenutku 2 s

Na slici 5.10 prikazan je maksimalni pomak konstrukcije koji se postiže u simulaciji, a strujanje se još razvija. Na slici 5.11 strujanje se razvilo u potpunosti dok konstrukcija još uvijek oscilira. Također se u simulaciji mogu primijetiti vrtlozi. Na slici 5.12 konstrukcija u potpunosti prigušena te je postigla konačni pomak.

### 5.3. Trodimenzionalni slučaj s vertikalnim stupom

Treći slučaj koji se analizira je trodimenzionalni slučaj vertikalnog stupa pravokutnog presjeka. Cilj ovog primjera je pokazati funkcioniranje implementiranog algoritma i na trodimenzionalnim mrežama kontrolnih volumena. Ovaj slučaj neće biti uspoređen s PRECICE. Diskretizacija konstrukcije je provedena trodimenzionalnim linearnim konačnim elementima. Broj konačnih elemenata je 160. Domena kontrolnih volumena je diskretizirana s 191892 kontrolnih volumena. Postavke simulacije navedene su u tablici 5.5.

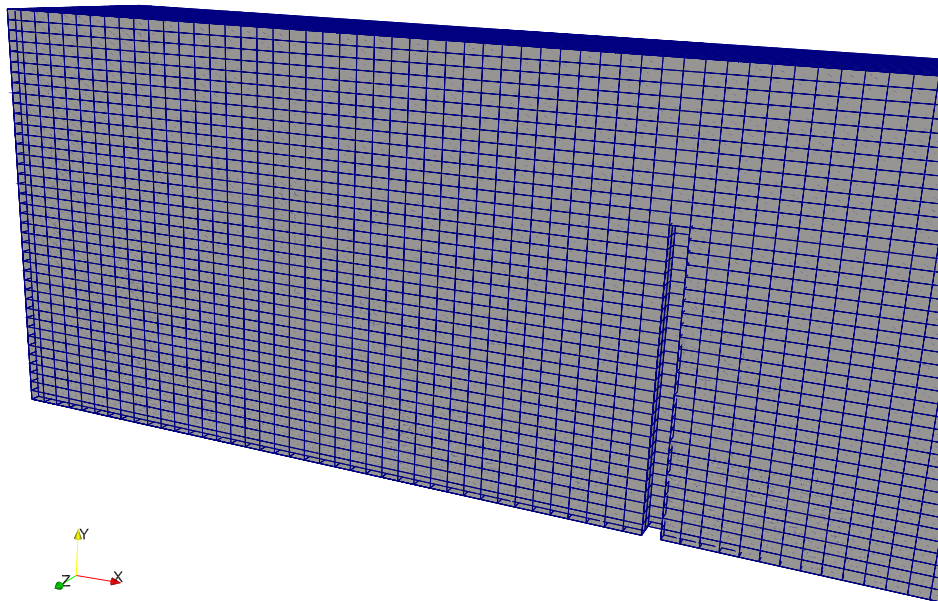
Tablica 5.5: Postavke simulacije trodimenzionalnog slučaj s vertikalnim stupom

$U_{\text{inf}}$	20 m/s
$E_S$	100 MPa
$\rho_F$	1.204 kg/m <sup>3</sup>
$\rho_S$	1000 kg/m <sup>3</sup>

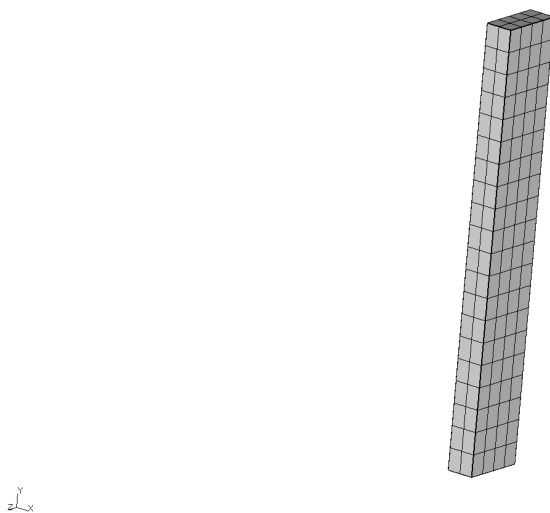
Tablica 5.6: Rubni uvjeti na mreži kontrolnih volumena

	Brzina	Tlak	Pomaci mreže
inlet	fixedValue (20 0 0)	zeroGradient	fixedValue (0 0 0)
outlet	zeroGradient	fixedValue 0	fixedValue (0 0 0)
lowerWall	fixedValue (0 0 0)	zeroGradient	fixedValue (0 0 0)
upperWall	symmetry	symmetry	symmetry
prepreka	movingWallVelocity	zeroGradient	fixedValue (0 0 0)
frontAndBack	symmetry	symmetry	symmetry

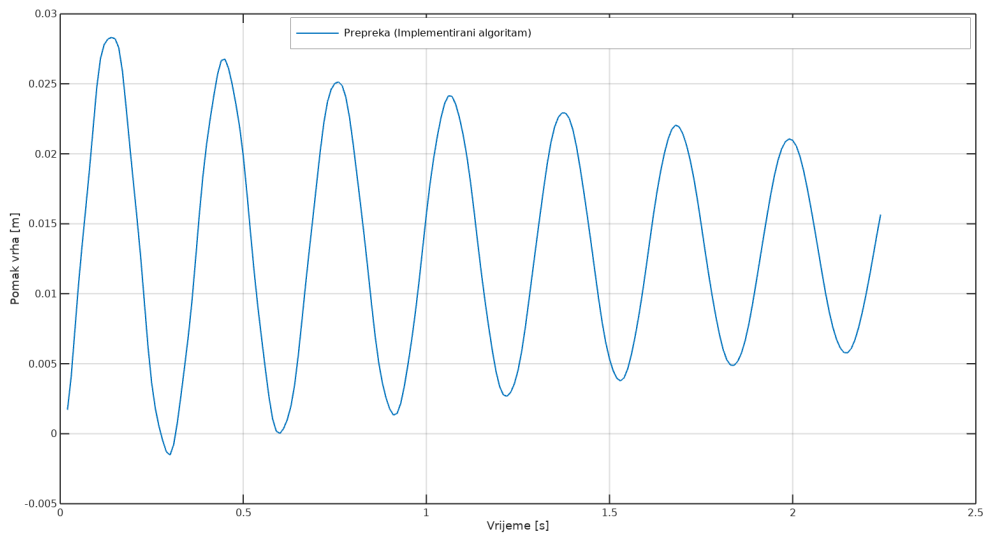
Na slici 5.13 prikazan je dio mreže kontrolnih volumena sa presjekom kako bi se dobio bolji uvid. Mreža konačnih elemenata prikazana je na slici 5.14. Rubni uvjet za vertikalni stup je uklještenje pri dnu.



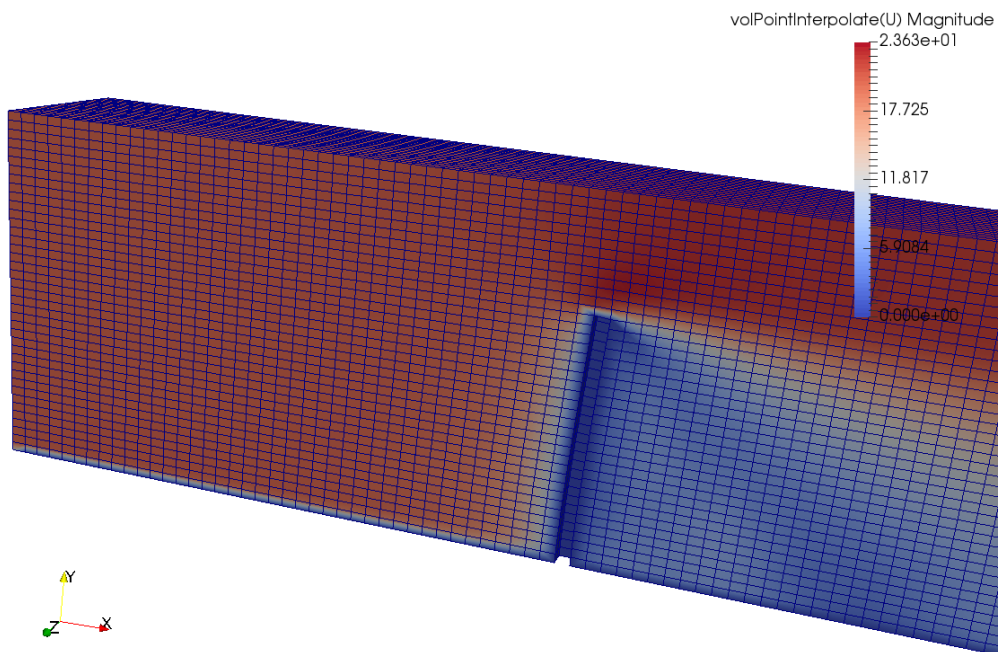
Slika 5.13: Mreža kontrolnih volumena za trodimenzionalni slučaj s vertikalnim stupom



Slika 5.14: Mreža konačnih elemenata za trodimenzionalni slučaj s vertikalnim stupom



Slika 5.15: Rezultati pomaka vrha za trodimenzionalni slučaj s vertikalnim stupom



Slika 5.16: Polje brzina u vremenskom trenutku 2.2 s - presjek

Sama simulacija zbog dugog trajanja nije provedena do kraja već je provedena do trenutka u kojem se može vidjeti nekoliko perioda gibanja vertikalnog stupa. Također se može primijetiti kako se maksimalni pomaci postepeno smanjuju što znači da je konstrukcija uspješno viskozno prigušena te da teži prema konačnom pomaku.

## 5.4. Polukrilo

Ovaj slučaj je jednostavno aluminijski polukrilo sa profilom NACA4412. Cilj ovog primjera je prikazati funkcioniranje algoritma za jedan složeni trodimenzionalni zrakovni primjer. Polukrilo je jednostavne geometrije s uzdužnicama, rebrima i dvije ramenjače. Diksretizirano je ljuskastim elementima te se sastoji od 3593 ljuskasta elementa. Dimenzije krila i konstrukcijskih elemenata krila prikazane su u tablici 5.7, a postavke simulacije u tablici 5.8.

Tablica 5.7: Dimenzije polukrila i konstrukcijskih elemenata

Raspon polukrila	3.5 m
Duljina tetive	1 m
Ramenjača	1 mm
Rebra	1 mm
Uzdužnice	1 mm

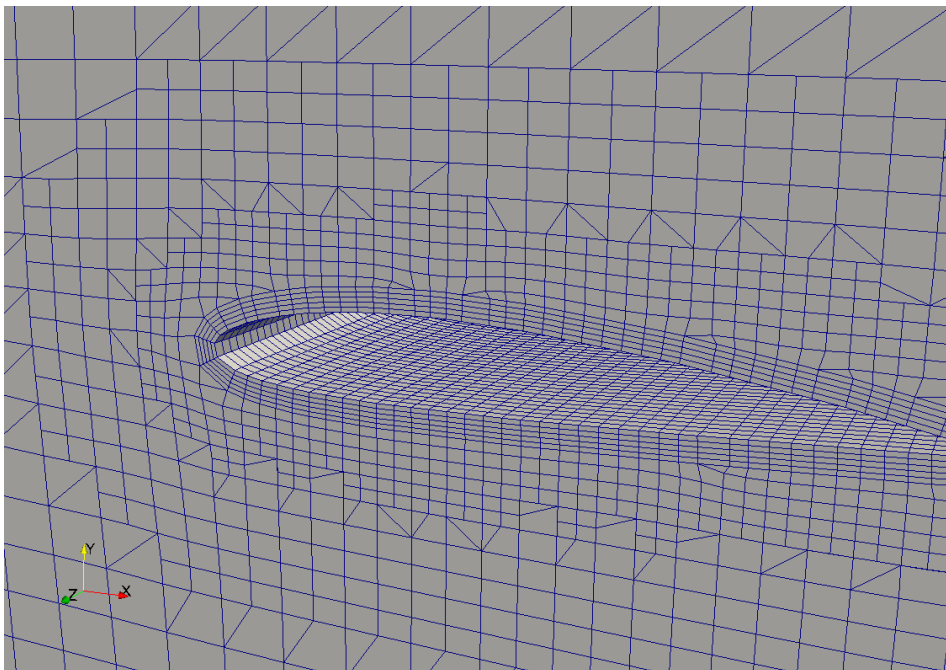
Tablica 5.8: Postavke simulacije za slučaj polukrilo

$U_{\text{inf}}$	60 m/s
$E_S$	75000 MPa
$\rho_F$	1.204 kg/m <sup>3</sup>
$\rho_S$	2750 kg/m <sup>3</sup>

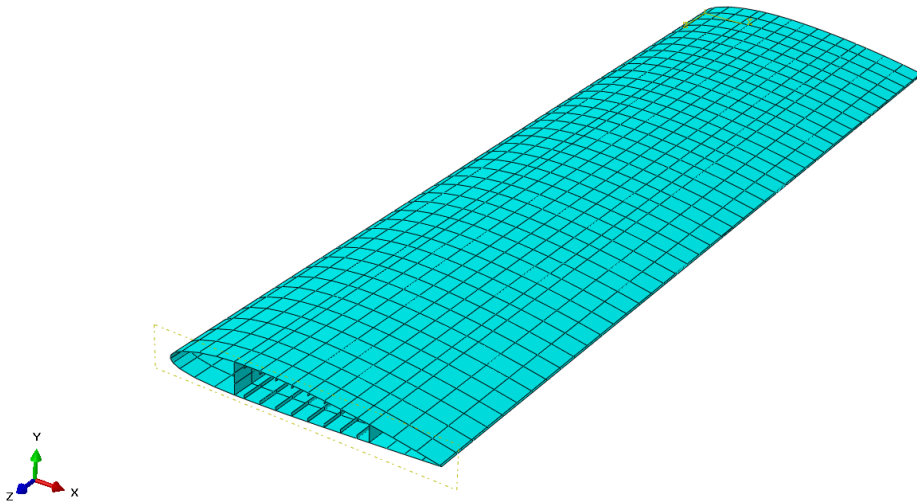
U nastavku su u tablici 5.9 navedeni rubni uvjeti za domenu kontrolnih volumena. Domena se sastoji od 270006 kontrolnih volumena. Diskretizacija je provedena tako da je polukrilo pod nulnim napadnim kutom. Simulacija je napravljena tako da se polukrilo odjednom nađe u struji fluida te se gleda odziv konstrukcije polukrila odnosno pomak vrha krila okomito na smjer strujanja fluida. Mreža je napravljena relativno grubo kako bi se što više ubrzala simulacija.

Tablica 5.9: Rubni uvjeti na mreži kontrolnih volumena za slučaj polukrila

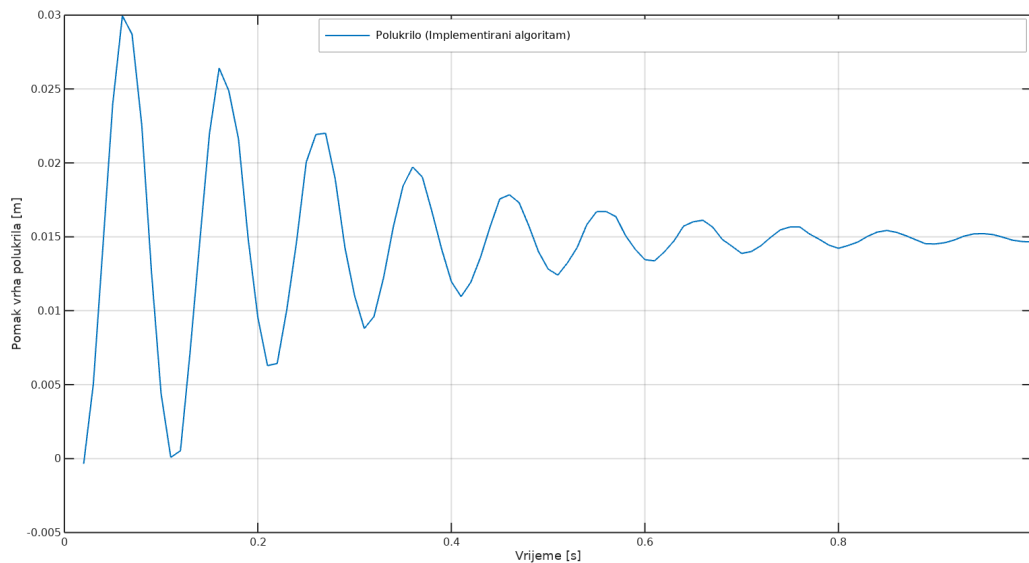
	Brzina	Tlak	Pomaci mreže
inlet	fixedValue (60 0 0)	zeroGradient	fixedValue (0 0 0)
outlet	zeroGradient	fixedValue 0	fixedValue (0 0 0)
lowerWall	symmetry	symmetry	symmetry
upperWall	symmetry	symmetry	symmetry
prepreka	movingWallVelocity	zeroGradient	fixedValue (0 0 0)
front	fixedValue (0 0 0)	symmetry	symmetry
back	symmetry	symmetry	symmetry



Slika 5.17: Mreža kontrolnih volumena za polukrilo - isječak



Slika 5.18: Mreža konačnih elemenata za polukrilo



Slika 5.19: Pomaci vrha polukrila u smjeru okomitom na strujanje

Polukrilo je započelo oscilacijsko gibanje koje se uslijed prigušenja fluida prigušilo te stabiliziralo na približno 15 milimetara. Ovakav ishod simulacije s obzirom na prethodne je bio očekivan. Ova simulacija, iako je ovaj slučaj inženjerski neinteresantan pokazala da



implementirani algoritam radi i kod ovakvih složenijih problema u skladu s očekivanjima što znači da se može koristiti za analizu aeroelastičnih pojava.

## 6 | Zaključak

U ovome radu implementirana je formulacija za proračun interakcije između fluida i elastične konstrukcije koristeći eksplicitni algoritam, zbog slabe veze između fluida i konstrukcije.

Na temelju provedenih analiza može se zaključiti kako je razvijeni algoritam postigao zadovoljavajuće rezultate te je otvorenog koda što je vrlo važno za inženjersku praksu. Samim time moguće su preinake u samom kodu te samostalno reprogramiranje koda. Na taj način je postignuta visoka fleksibilnost razvijenog algoritma. Implementirana formulacija pokazuje relativno dobro slaganje s temeljito provjerenom bibliotekom PRECICE.

Jedan od prijedloga za daljnji rad je istražiti uzroke manjih odstupanja rezultata sa uspoređenim algoritmom u prva dva slučaja. Daljnji prijedlozi su efikasnija implementacija u C++. Potencijal implementiranog algoritma je proširenje i na implicitno povezivanje, što bi omogućilo i simulacije problema u kojima su fluid i konstrukcija u snažnoj vezi.

# Literatura

- [1] [www.calculix.de](http://www.calculix.de).
- [2] <https://github.com/Unofficial-Extend-Project-Mirror/foam-extend-foam-extend-4.0>.
- [3] Dan Negrut, Gisli Ottarsson, Rajvi Rampalli, and Anthony Sajdak. *On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential-Algebraic Equations of Multibody Dynamics*. ASME 2005 International Design Engineering Technical Conferences and 5th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Long Beach, USA, 2006.
- [4] Pongpat Thavornpattanapong. Performance of fluid-structure interaction based on analytical and computational techniques. Master's thesis, RMIT University, 2011.
- [5] Ž. Tuković. *Metoda kontrolnih volumena na domenama promjenjivog oblika*. PhD thesis, Fakultet strojarstva i brodogradnje, 2005.
- [6] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Tokyo, 2002.
- [7] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics*. Springer International Publishing Switzerland, Springer Cham Heidelberg New York Dordrecht London, 2016.

- [8] J. Degroote. *Development of Algorithms for the Partitioned Simulation of Strongly Coupled Fluid-Structure Interaction Problems*. PhD thesis, Ghent University, 2010.
- [9] Pongpat Thavornpattanapong. Fluid-structure interaction (fsi) case study of a cantilever using openfoam and deal.ii with application to viv. Master's thesis, Lunds Institute of Technology, 2009.
- [10] Augustina De Boer. *Computational fluid-structure interaction*. PhD thesis, Technische Universiteit Delft, 2008.
- [11] T. Richter. *Fluid-structure Interactions*. Springer International Publishing AG, Gewerbestrasse 11, 6330 Cham, Switzerland, 2017.
- [12] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems part ii: energy transfer analysis and three-dimensional applications. *Comput. Methods Appl. Mech. Engrg.* 190 (2001), 1999.
- [13] Pablo Mosquera Michaelsen. *A Fluid-Structure-Interaction simulation tool for application in rotating machinery*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2015.
- [14] K. Scheufele. Robust quasi-newton methods for partitioned fluid-structure simulations. Master's thesis, University of Stuttgart, 2015.
- [15] M. Kerrisk. *The Linux Programming Interface*. William Pollock, 38 Ringold Street, San Francisco, CA 94103, 2010.
- [16] [www.gnu.org/software/libc/manual/html\\_node/Sockets.html](http://www.gnu.org/software/libc/manual/html_node/Sockets.html).
- [17] B. Gatzhammer. *Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions*. PhD thesis, Technische Universität München, 2014.
- [18] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 2000.
- [19] Z. Majdisova and V. Skala. A radial basis function approximation for large datasets. *SIGRAD*, 2016.

- [20] S. Kopysov, I. Kuzmin, A. Novikov, N. Nedozhogin, and L. Tonkov. Radial basis function for mesh to mesh interpolation in parallel solving fluid-structure interaction problem. *Russian Supercomputing Days*, 2017.
- [21] [http://www.eigen.tuxfamily.org/index.php?title=Main\\_Page](http://www.eigen.tuxfamily.org/index.php?title=Main_Page).
- [22] M. Mongillo. Choosing basis functions and shape parameters for radial basis function methods. 2011.
- [23] A. Bogaers. *Efficient and robust partitioned solution schemes for fluid-structure interactions*. PhD thesis, University of Cape Town, 2015.
- [24] [www.precice.org](http://www.precice.org).