Numeričke metode za rješavanje Navier-Stokes jednadžbe

Marin Vilović
Mentor: izv.prof.dr.sc. Davor Horvatić
Fizički odsjek, Prirodoslovno-matematički fakultet, Bijenička 32, Zagreb

U ovom radu su predstavljenje tri popularne numeričke metode za rješavanje Navier-Stokes jednadžbe. Metode su detaljno opisane te primjenjene na problem u računalnoj dinamici fluida. Simulacije su izvedene koristeći C++ softverski paket OpenFOAM i vizualno analizirane koristeći ParaView.

26. siječnja 2020.

1 Uvod

Navier-Stokes jednadžba, nazvana po fizičarima Claude-Louis Navieru i George Gabriel Stokesu, opisuje gibanje viskoznih fluida. Jednadžba se zasniva na pretpostavci da je fluid, na određenoj promatranoj skali, kontinuum umjesto skupina diskretnih čestica. Izvodi se primjenom jednadžbe kontinuiteta na sačuvanje mase, količine gibanja i energije.

Navier-Stokes jednadžba je korisna pri opisu fizikalnih procesa u problemima koji su interesantni u znanosti i inženjerstvu. Pomoću nje možemo modelirati vremenske prilike, promatrati gibanje vode u cijevi ili zraka oko krila zrakoplova. Jednadžba je također zanimljiva u matematičkom pogledu. Iako postoji široka primjena same jednadžbe, još nije dokazano da uvijek postoji glatko rješenje u tri dimenzije. O težini problema govori i činjenica da je *Clay Mathematics Insitute* nazvao ovaj problem "jedan od sedam najvažnijih otvorenih problema u matematici" te ponudio nagradu od milijun dolara za rješenje ili kontraprimjer spomenutog problema[1].

Dinamika fluida je grana fizike koja je, uz zakone termodinamike, opisana Navier-Stokes jednadžbom i jednadžbom kontinuiteta. Analitičko rješenje spomenutih jednadžbi je poznato samo za jednostavne slučajeve i to uz zanemarivanje određenih članova samih jednadžbi. Stoga, u rješavanju Navier-Stokes jednadžbe veliku ulogu imaju numeričke metode.

U ovom seminaru se bavimo podgranom dinamike fluida koja se zove računalna dinamika fluida (eng. *Computational Fluid Dynamics*). Predmet promatranja u našem slučaju su nestlačivi fluidi Newtonovog tipa.

Navier-Stokes jednadžba i jednadžba kontinuiteta za takav fluid dane su izrazom:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} - \nu \nabla^2 \boldsymbol{u} = -\nabla p + \boldsymbol{f} \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2}$$

gdje $\boldsymbol{\nu}$ predstavlja kinematičku viskoznost. Uobičajena Navier-Stokes jednadžba podijeljena je sa gustoćom fluida ρ zbog bolje preglednosti. Stoga, u jednadžbi (1) varijabla p predstavlja tlak podijeljen sa gustoćom fluida. Radi jednostavnosti, u daljnjem tekstu varijablu p nazivamo tlak.

Prvi član u jednadžbi (1) sa lijeve strane predstavlja derivaciju brzine $\boldsymbol{u}$. Drugi član sa lijeve strane $(\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u}$ nazivamo konvekcijskim članom, a treći član $\nu \nabla^2 \boldsymbol{u}$ difuzijskim članom. Članovi sa desne strane su redom gradijent tlaka ∇p i sila po jedinici mase $\boldsymbol{f}$.

Rješavanje Navier-Stokes jednadžbe se svodi, uz prikladne rubne uvjete, na pronalazak rješenja za polja brzine i tlaka koja su međusobno povezana. U ovom radu će biti predstavljene tri popularne numeričke metode za rješavanje Navier-Stokes jednadžbe. Svaka od njih se odnosi prema problemu polja brzine i tlaka na različiti način. Stoga, očekujemo da će postojati razlike u rezultatima. S obzirom na to da u problemima računalne dinamike fluida ne postoji egzaktno ili točno rješenje, već je to uvijek aproksimacija do određene mjere, rezultati će primarno biti uspoređeni vizualno. Također treba obratiti pažnju na jedan od najvažnijih čimbenika kod računalnih simulacija, a to je vrijeme izvršavanja simulacije.

2 Metodologija

Predmet promatranja ovog rada je laminarni tok nestlačivog fluida Newtonovog tipa. Prije primjene numeričkih metoda, potrebno je generirati mrežu (eng. *mesh*) na geometriji odabranog problema. Drugim riječima, vršimo diskretizaciju prostorne domene i dijelimo je na konačan broj kontrolnih volumena. Upravo se ti volumeni koriste kao diskretne lokalne aproksimacije kod primjene numeričkih metoda.

Za rješavanje problema lamiarnog toka nestlačivog fluida često se koristi metoda bazirana na PISO (eng. *Pressure-Implicit with Splitting of Operators*) algoritmu. Radi se o tranzijentnoj metodi koja se koristi kada nas osim konvergiranog rezultata, zanima i način na koji je do njega došlo. Drugim riječima, takvom metodom dobijemo potpun vremenski opis problema, od početnog trenutka do konvergencije. S druge strane, vrlo popularna metoda za traženje samo stacionarnog tj. konvergiranog rješenja bazirana je na SIMPLE (eng. *Semi-Implicit Method for Pressure Linked Equations*) algoritmu. Kao treću metodu koju ćemo obraditi, odabrana je metoda bazirana na PIMPLE algoritmu. Radi se o algoritmu koji je nastao kao kombinacija spomenutih PISO i SIMPLE algoritama. PIMPLE metoda je također tranzijentna, ali nakon svakog koraka u računu traži stacionarno rješenje. Takav način uvelike ubrzava vrijeme simulacije u odnosu na PISO metodu, ali zaostaje u detaljnosti opisa samog toka.

2.1 PISO algoritam

PISO metoda je tranzijenta metoda koja najbolje radi pri malim vremenskim koracima. Prvi korak PISO metode je rješavanje Navier-Stokes jednadžbe (1) s vrijednostima tlaka iz prethodnog vremenskog koraka. Nadalje, potrebno je zadovoljiti i jednadžbu kontinuiteta (2). To se postiže supstitucijom izraza za površinski protok, izražen iz pretpostavljenog polja brzine, u jednadžbu kontinuiteta. Dobivena jednadžba je jednadžba tlaka te njeno rješenje daje pretpostavljeno polje tlaka i korekciju polja brzine i površinskog protoka. Tada se izvede druga jednadžba za korekciju tlaka koja kao rješenje daje ispravljeno polje tlaka te drugu korekciju brzine i površinski protoka. Navedeni koraci se ponavljaju do konvergencije[2].

Počinjemo od Navier-Stokesovih jednadžbi, zapisanih u malo drugačijem obliku u odnosu na uvod:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) - \nabla \cdot (\nu \nabla \boldsymbol{u}) = -\nabla p \tag{3}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{4}$$

Nelinarnost u konvekcijskom članu $\nabla \cdot (\boldsymbol{u}\boldsymbol{u})$ rješavamo koristeći iterativnu tehniku za koju vrijedi izraz:

$$\nabla \cdot (\boldsymbol{u}\boldsymbol{u}) \approx \nabla \cdot (\boldsymbol{u}^{\boldsymbol{o}}\boldsymbol{u}^{\boldsymbol{n}}), \tag{5}$$

gdje $\boldsymbol{u}^{\boldsymbol{o}}$ predstavlja trenutno rješenje polja brzine, a $\boldsymbol{u}^{\boldsymbol{n}}$ novo rješenje. Treba naglasiti da prethodni izraz vrijedi samo u slučaju kada je vremenski korak dovoljno male vrijednosti. S obzirom da se radi o nestlačivom fluidu, jednadžba za tlak nepostoji. Upravo tu se pozivamo na jednadžbu kontinuiteta, preko koje ćemo izvesti jednadžbu za tlak u daljnjem tekstu.

U prvom koraku simulacije provodimo diskretizaciju Navier-Stokes jednadžbe (3) metodom konačnih volumena uz zadržavanje člana gradijenta tlaka:

$$a_P^{\boldsymbol{u}} \boldsymbol{u}_P + \sum_N a_N^{\boldsymbol{u}} \boldsymbol{u}_N = \boldsymbol{r} - \nabla p, \tag{6}$$

gdje su a_P dijagonanalni koeficijenti matrice koji predstavljaju određenu ćeliju, a_N koeficijenti matrice uz dijagonalu koji predstavljaju susjede promatrane ćelije i $\boldsymbol{r}$ član koji sadrži sve članove koji se mogu eksplicitno izračunati. Rješavamo jednadžbu (6) s vrijednostima tlaka iz prošlog koraka te kao rezultat dobijemo polje brzine. Uvodimo opreator $\boldsymbol{H}(\boldsymbol{u})$

$$\boldsymbol{H}(\boldsymbol{u}) = \boldsymbol{r} - \sum_N a_N^{\boldsymbol{u}} \boldsymbol{u}_N \tag{7}$$

tako da vrijedi

$$\boldsymbol{u}_P = \left(a_P^{\boldsymbol{u}}\right)^{-1} \left(\boldsymbol{H}(\boldsymbol{u}) - \nabla p\right) \tag{8}$$

Supstitucijom izraza (8) u jednadžbu kontinuiteta (4) dobije se jednadžba za tlak nestlačivog fluida:

$$\nabla \cdot \left[(a_P^{\boldsymbol{u}})^{-1} \nabla p\right] = \nabla \cdot \left[(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{H}(\boldsymbol{u})\right] \tag{9}$$

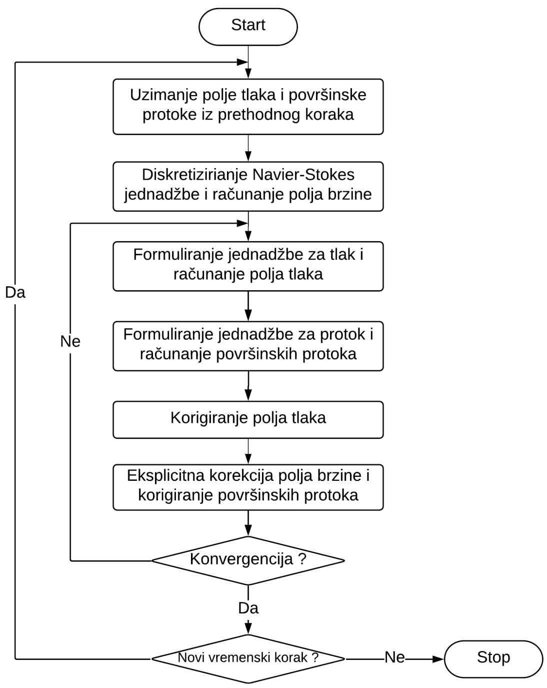
Sada rješavamo jednadžbu tlaka (9) koristeći već izračunato polje brzine. Nadalje, uvodimo diskretizaciju jednadžbe kontinuiteta s ciljem izvoda jednadžbe za površinski protok F:

$$\nabla \cdot \boldsymbol{u} = \sum_f F = \sum_f \boldsymbol{s}_f \cdot \boldsymbol{u} \tag{10}$$

gdje $\boldsymbol{s}_f$ ima ulogu vektora smjera. Supstitucijom izraza za polje brzine (8) u prijašnju jednadžbu dobije se:

$$F = -\left(a_P^{\boldsymbol{u}}\right)^{-1} \boldsymbol{s}_f \cdot \nabla p + \left(a_P^{\boldsymbol{u}}\right)^{-1} \boldsymbol{s}_f \cdot \boldsymbol{H}(\boldsymbol{u}) \tag{11}$$

gdje možemo primjetiti da prvi član na lijevoj strani jednadžbe (11) odgovara desnoj strani jednadžbe (9). Izračunom površinskih protoka rekonstruira se polje brzine te korigira oprator $\boldsymbol{H}(\boldsymbol{u})$. Sada koristimo dobiveni tlak i korigiranu vrijednost operatora $\boldsymbol{H(u)}$ kako bi eksplicitno korigirali polje brzine dano jednadžbom (8). Iz toga se računaju novi površinski protoci te se svi navedeni postupci od jednadžbe (6) ponavljaju do konvergencije[3][4]. Cijela petlja PISO algoritma prikazana je u obliku dijagrama (Slika 1).



Slika 1: Dijagram petlje PISO algoritma

2.2 SIMPLE algoritam

SIMPLE metoda je stacionarna metoda koja radi pri velikim vremenskim koracima i čiji je cilj brz dolazak do konvergentnog rješenja. S obzirom da nemamo vremensku derivaciju u jednadžbi (3), potrebno je uvesti podrelaksaciju traženih polja kako bi postigli stabilnost i konvergenciju.[3]

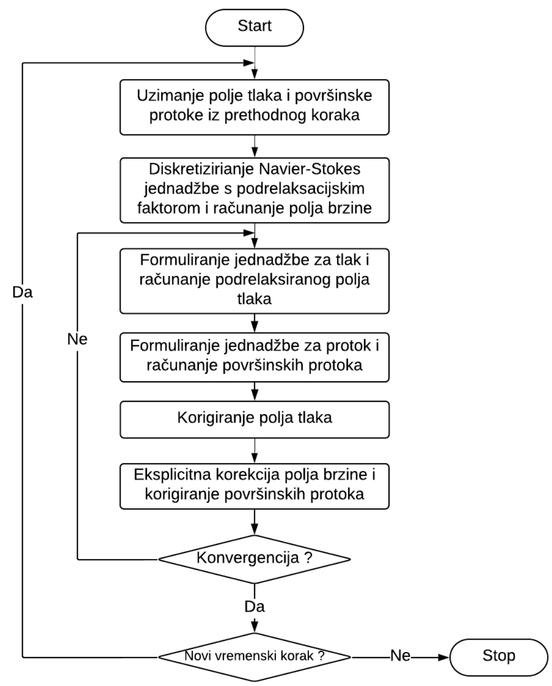
U prvom koraku SIMPLE algoritma uvodimo implicitno podrelaksiranje diskretizirane Navier-Stokes jednadžbe:

$$\frac{a_P^{\boldsymbol{u}}}{r_u}\boldsymbol{u}_P^{\boldsymbol{n}} + \sum_N a_N^{\boldsymbol{u}}\boldsymbol{u}_N = \boldsymbol{r} - \nabla p + \frac{1-r_u}{r_u}\boldsymbol{u}_P^{\boldsymbol{o}} \tag{12}$$

gdje je r_u podrelaksacijski faktor za polje brzine. Rješavanjem jednadžbe (12) izračuna se polje brzine iz kojeg računamo površinske protoke. Nadalje, izračunamo operator $\boldsymbol{H(u)}$ (7), što omogućava formuliranje jednadžbe za tlak (9). Rješavanjem jednadžbe za tlak dobije se polje tlaka. Na dobivenom polju tlaka provodimo podrelaksaciju danu izrazom:

$$p^n = p^o + r_p \cdot (p^p - p^o), \tag{13}$$

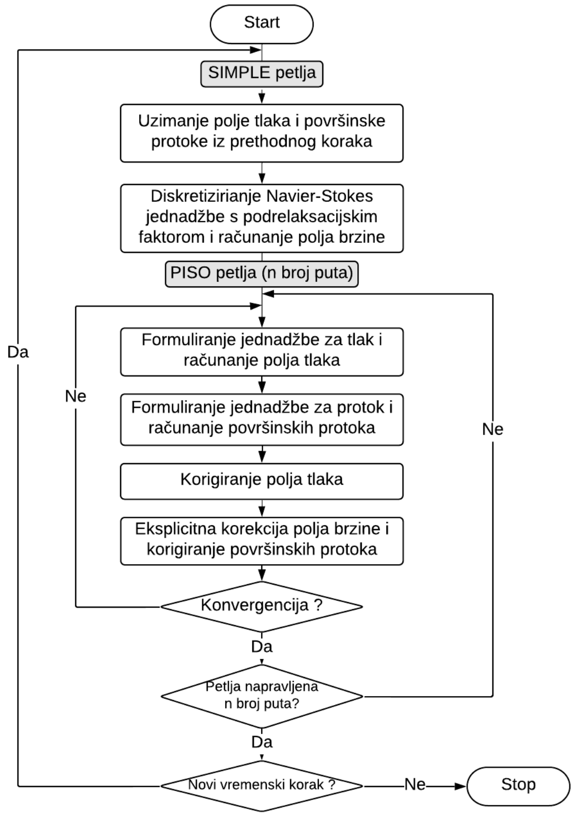
gdje je r_p podrelaksacijski faktor za polje tlaka, p^p rješenje jednadžbe tlaka, p^o tlak iz prethodnog koraka i p^n podrelaksirani tlak. Kao i kod PISO metode, pomoću dobivenog tlaka korigiramo polje brzine i površinske protoke. Svi navedeni postupci se ponavljaju do konvergencije[3][5]. Cijela petlja SIMPLE algoritma prikazana je u obliku dijagrama (Slika 2).



Slika 2: Dijagram petlje SIMPLE algoritma

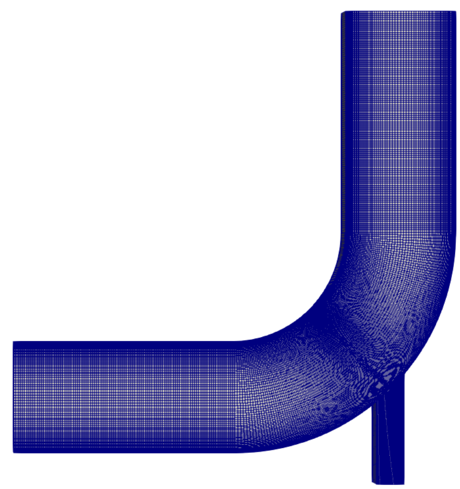
2.3 PIMPLE algoritam

PIMPLE algoritam je jedan od najčešće korištenih algoritama za tranzijente probleme. Nastao je kao kombinacija PISO i SIMPLE algoritama. Prednost je što se algoritam može koristiti za velike vremenske korake, a i dalje ima tranzijentnu karakteristiku. Unutar jednog vremenskog koraka, stacionarno rješenje se traži primjenom podrelaksacije. Nakon pronalaska stacionarnog rješenja ulazimo u petlju PISO petlju koja se ponavlja određeni broj puta, definirano u postavkama od strane korisnika. Nakon što PISO petlja potrvdi da su svi eksplicitni dijelovi jednadžbe konvergirali, krećemo na sljedeći vremenski korak. Cijeli postupak se ponavlja do konvergencije[6]. Cijela petlja PIMPLE algoritma prikazana je u obliku dijagrama (Slika 3).



Slika 3: Dijagram petlje PIMPLE algoritma

Slika 4: Geometrija odabranog 2D problema



Slika 5: Generirana mreža na odabranoj geometriji

3 Problem i priprema simulacije

OpenFOAM (eng. *Open Source Field Operation and Manipulation*) je C++ softverski paket otvorenog koda za rješavanje kompleksnih problema u mehanici fluida. Uz veliki broj implementiranih numeričkih metoda, OpenFOAM također sadrži alate za generiranje potrebnih mreža kontrolnih volumena. Za obradu rezultata korišten je ParaView, poznati alat otvorenog koda za vizualizaciju. Simulacije se pokreću iz terminala Linux operativnog sustava. Potrebno je u istoj mapi kroz tekstualne datoteke definirati rubne uvjete, generirati mrežu i odrediti postavke simulacije kao što su vrijeme koraka i vrijeme uzorkovanja[7][8].

Odabrani problem za analizu razlika numeričkih metoda je 2D geometrija oblika prikazanog grafički (Slika 4). S obzirom da se radi o geometriji koja je dobivena u sklopu OpenFOAM softverskog paketa, zasigurno će dati valjani rezultat. Kao što je spomenuto u prethodnom poglavu, prije početka simulacije potrebno je generirati mrežu na geometriji. Mreža se također generira unutar OpenFOAM paketa. S ciljem da rezultati budu što točniji, mreža je vrlo detaljna i sadrži otprilike 35 do 40 tisuća ćelija u oba smjera (Slika 5).

Uz odabir geometrije i generiranje mreže, potrebno je postaviti rubne uvjete domene. Dirichletov i Neumannov su dva tipa rubnih uvjeta. Dirichletov propisuje vrijednost parcijalne diferencijalne jednadžbe, a Neumannov vrijednost nultog gradijenta. U slučaju generaliziranog Neumannovog uvjeta, propisuje se proizvoljna vrijednost gradijenta varijable u smjeru normale na površinu granice. U računalnoj dinamici fluida najčešći rubni uvjeti su ulazna granica, izlazna granica i zid. Ulazna granica, na kojoj zadajemo Dirichletov uvjet, određuje brzinu nastrujavanja ulaznog fluida. Na izlaznoj granici, na kojoj zadajemo Neumannov uvjet, za tlak se postavlja konstanta vrijednost. Zid ima ulogu nepropusne granice, gdje kod viskoznog strujanja fluid ima brzinu jednaku brzini napredovanja zida.

Za odabranu geometriju odabrane su dvije ulazne granice brzine. Radi se o ulazu sa lijeve i donje strane. Sa lijeve strane odabrano je da u smjeru normale na površinu granicu ulazi odabrani fluid brzinom od $1\,\mathrm{m\,s^{-1}}$. Sa donje strane, na isti način ulazi odabrani fluid brzinom od $3\,\mathrm{m\,s^{-1}}$. Na izlaznoj granici, koja se nalazi na gornjoj strani geometrije, odabrana je konstanta vrijednost za tlak. S ciljem lakše analize rezultata, odabrana referenta vrijednost tlaka na granici je 0 Pa. Ostaje odabrati rubni uvjet za zid. S obzirom da želimo promatrati tok unutar cijevi koja se ne giba u prostoru, brzina fluida uz stijenku je jednaka nuli. Svi navedeni rubni uvjeti su prikazani i tablično (Tablica 1).

Granica domene	Varijabla	Vrijednost varijable
Lijevi ulaz	$\boldsymbol{u}$	$1\,\mathrm{m\,s^{-1}}$
Donji ulaz	$\boldsymbol{u}$	$3\,\mathrm{m\,s^{-1}}$
Gornji izlaz	p	0 Pa
Stijenka	$\boldsymbol{u}$	$0\,\mathrm{m\,s^{-1}}$

Tablica 1. Rubni uvjeti simulacije

Karakteristike fluida se definiraju preko kinematičke viskoznosti. Za naše simulacije odabrana je kinematička viskoznost koja odgovara ulju. Za provedbu istih simulacija za vodu ili zrak, koji su podložniji nepredvidivom i kompliciranom gibanju, potrebna je mnogo veća procesorska snaga.

Nakon pripreme geometrije, mreže i rubnih uvjeta za simulaciju potrebno je odabrati prikladne numeričke solvere koji odgovaraju našim metodama. S obzirom da se radi o popularnim algoritmima, implementirani su u OpenFOAM paketu. Numerički solver koji odgovara PISO algoritmu se poziva naredbom *icoFoam*, solver koji odgovara SIMPLE algoritmu poziva se naredbom *simpleFoam* i solver koji odgovara PIMPLE algoritmu poziva se naredbom *pimpleFoam*.

4 Rezultati i analiza

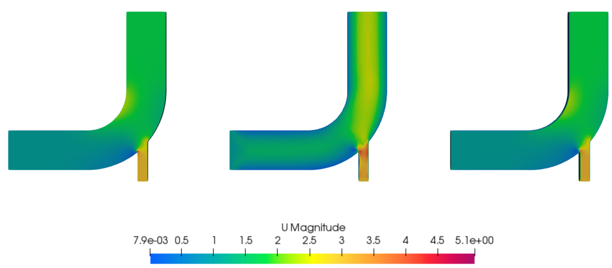
Nakon provedbe simulacije, potrebno je usporediti numeričke metode. S obzirom na to da u računalnoj dinamici fluida ne postoji egzaktno rješenje, rezultati će biti uspoređeni vizualno i u vidu vremena potrebnog za provedbu same simulacije.

Očekujemo da je za konvergenciju rezultata PISO metode potrebno najviše vremena, a za konvergenciju rezultata SIMPLE metode najmanje vremena. Rezultati potvrđuju očekivanja i vrijeme potrebno za konvergenciju pojedine metode je prikazno tablično (Tablica 2). Iako samo vrijeme izvršavanja ovisi o dostupnoj računalnoj snazi, relativna usporedba je valjana.

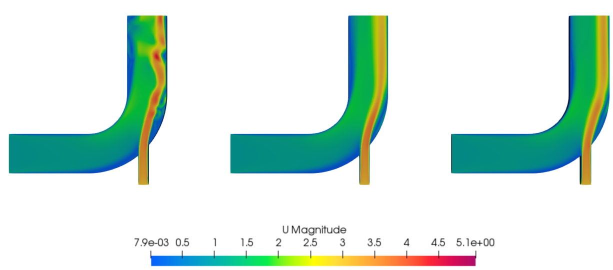
Metoda	Vrijeme izvršavanja simulacije
PISO	20742.85 s ≈ 5hr 54min
SIMPLE	54.12 s ≈ 1min
PIMPLE	2070.56 s ≈ 35min

Tablica 2. Vrijeme izvršavanja simulacije za pojedinu metodu

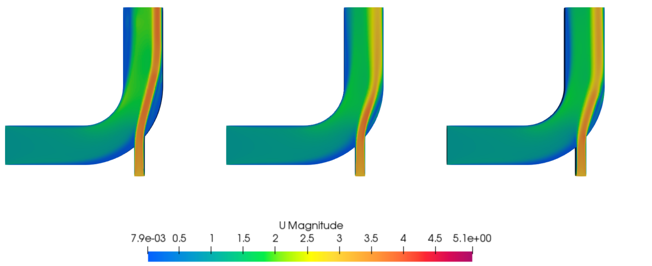
Rezultate polja brzina ćemo prikazati vizualno u 3 vremenska koraka simulacije (Slika 6, Slika 7, Slika 8). SIMPLE metoda je stacionarna metoda te nam nakon nekoliko vremenskih koraka daje konvergirano rješenje. S druge strane, PISO i PIMPLE su tranzijente metode za koje je potrebno više vremenskih koraka. Uz početni i zadnji vremenski korak, odabiremo približno polovicu ukupnog simulacijskog vremena za PISO metodu koja konvergira nakon 220 sekundi simulacijskog vremena.



Slika 6: Polje brzine za vremenski korak $t = 0.1$ s. Metode su slijeva nadesno PISO, SIMPLE i PIMPLE algoritam.



Slika 7: Polje brzine za vremenski korak $t = 100$ s. Metode su slijeva nadesno PISO, SIMPLE i PIMPLE algoritam.



Slika 8: Polje brzine za vremenski korak $t = 220$ s. Metode su slijeva nadesno PISO, SIMPLE i PIMPLE algoritam.

Vizualnim pregledom prikazana 3 vremenska koraka, potvrđujemo prijašnje pretpostavke. PISO algoritam nudi najdetaljniji opis toka te posljedično zahtjeva najduže simulacijsko vrijeme od 220 s za konvergenciju. Naime, zbog malih vremenskih koraka tok postaje puno detaljniji kao što se može i vidjeti za vremenski korak $t = 100$ s (Slika 7). SIMPLE algoritam, kao stacionarni algoritam, daje konvergirano rješenje za samo 5 s simulacijskog vremena. PIMPLE algoritam zaobilazi vrlo detaljan opis toka PISO algoritma te konvergira za 125 s simulacijskog vremena. Konvergirani rezultati nisu potpuno isti i postoje vizualne razlike koje se mogu primjetiti (Slika 8). S obzirom da ne postoji egzaktno rješenje ovakvog problema, ne postoji ni točan rezultat ili jedinstvena metoda za primjenu.

5 Zaključak

U ovom radu su predstavljene tri popularne numeričke metode za rješavanje Navier-Stokes jednadžbe. Metode su detaljno opisane, obrađeni su diskretizirani fizikalni izrazi koji se koriste u samom računanju i prikazani su dijagrami toka pojedinog algoritma. Metode su primjenjene na problem laminarnog toka nestlačivog fluida Newtonovog tipa u računalnoj dinamici fluida. Nakon odabira geometrije i generiranja mreže, određeni su rubni uvjeti simulacije i definirane karakteristike fluida. Simulacije su provedene, rezultati uspoređeni vizualno i u vidu vremena izvršavanja simulacije. Dobiveni rezultati simulacija su usklađeni sa očekivanjima. PISO algoritam daje najdetaljniji opis toka, ali zahtijeva 5 sati i 54 minute za konvergenciju toka. S druge strane, SIMPLE algoritam u vrlo kratkom vremenskom roku od jedne minute daje konvergirani rezultat. PIMPLE algoritam koristi kombinaciju prethodno navedenih metoda što se odrazilo i na vrijeme izvršavanja simulacije, koje iznosi 35 minuta. Kod konvergiranih rezultata se mogu primjetiti vizualne razlike. S obzirom na prirodu problema nije moguće odabrati ispravno rješenje, već ih samo usporediti što je i bio cilj ovog rada.

Literatura

[1] `http://www.claymath.org/millennium-problems/navier-stokes-equation`

[2] Bressloff, Neil. (2001). A parallel pressure implicit splitting of operators algorithm applied to flow at all speeds. International Journal for Numerical Methods in Fluids. 36. 497 - 518.

[3] Ferziger, Joel H., Peric, Milovan. (2002). Computational Methods For Fluid Dynamics.

[4] Nillson, Håkan. (2017). Proceedings of CFD with OpenSource Software.

[5] Jang, D.S., Jetli, R., Acharya, S. (1986). Comparison of the PISO, SIMPLER and SIMPLEC algorithms for the treatment of the pressure-velocity coupling in steady flow problems, Numerical Heat Transfer. International Journal of Computation and Methodology, 10:3, 209-228

[6] Holzmann, Tobias. (2019). Mathematics, Numerics, Derivations and OpenFOAM®.

[7] `https://www.openfoam.com/`

[8] `https://www.paraview.org/overview/`