Numeričke metode za rješavanje Navier-Stokes jednadžbe

Marin Vilović

Mentor: izv.prof.dr.sc. Davor Horvatić

Uvod

- Navier-Stokes jednadžba opisuje gibanje viskoznih fluida
- Zasniva se na pretpostavci da je fluid kontinuum umjesto skupina diskretnih čestica
- Izvodi se primjenom jednadžbe kontinuiteta na sačuvanje mase, momenta i energije
- Široka primjena jednadžbe u znanosti i inženjerstvu
- Još nije dokazano da uvijek postoji glatko rješenje u tri dimenzije
- Jedan od sedam milenijskih problema u matematici

Dinamika fluida

- Grana fizike opisana Navier-Stokes jednadžbom i jednadžbom kontinuiteta
- Analitičko rješenje spomenutih jednadžbi poznato samo za jednostavne slučajeve uz zanemarivanje određenih članova jednadžbi
- U rješavanju veliku ulogu imaju numeričke metode

Računalna dinamika fluida

- eng. *Computational Fluid Dynamics*
- Predmet promatranja ovog radasu nestlačivi fluidi Newtonovog tipa
- Navier-Stokes jednadžba i jednadžba kontinuiteta za takav fluid su dane izrazom:

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

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

- $\boldsymbol{\nu}$ predstavlja kinematičku viskoznost

Navier-Stokes jednadžba

Vremenska derivacija polja brzine

Difuzijski član

Sila po jedinici mase

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

Konvekcijski član

Gradijent tlaka

Rješavanje Navier-Stokes jednadžbe

- Svodi se na pronalazak rješenja za polja brzine i tlaka
- Potrebno imati prikladne rubne uvjete
- Ne postoji egzaktno ili točno rješenje, već je to uvijek aproksimacija do određene mjere

Metodologija

- Predmet promatranja 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

Numeričke metode

- PISO algoritam – tranzijentna metoda
- SIMPLE algoritam – stacionarna metoda
- PIMPLE algoritam – tranzijenta metoda, ali nastaje kao kombinacija prethodne dvije

PISO algoritam

- eng. *Pressure-Implicit with Splitting of Operators algorithm*
- Metoda koja najbolje radi pri malim vremenskim koracima
- Koristi se kada nas osim konvergiranog rezultata, zanima i način na koji je do njega došlo
- Drugim riječima, dobijemo potpuni vremenski opis problema
- Nelinearnost u konvekcijskom se rješava koristeći iterativnu tehniku za koju vrijedi:

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

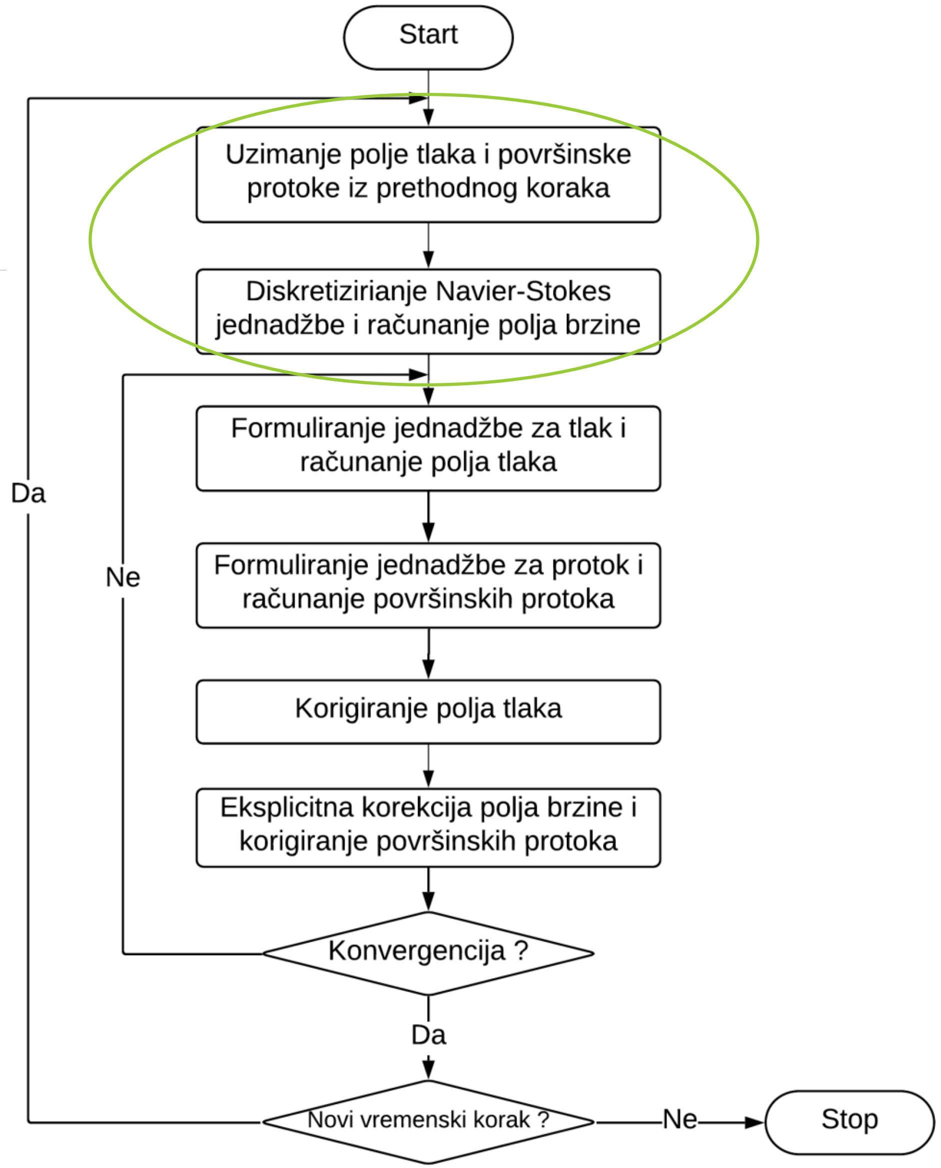
  - $\boldsymbol{u}^o$ - trenutno rješenje polja brzine
  - $\boldsymbol{u}^n$ - novo rješenje polja brzine

PISO algoritam

- Algoritam počinje preuzimanjem polja tlaka i površinskih protoka iz prethodnog koraka
- Provodimo diskretizaciju N-S jednadžbe metodom konačnih volumena:

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

  - a_P - dijagonalni koeficijenti matrice koji predstavljaju određenu ćeliju
  - a_N - koeficijenti matrice uz dijagonalu koji predstavljaju susjede promatrane ćeliju
  - $\boldsymbol{r}$ - svi članovi N-S jednadžbe koji se mogu eksplicitno izračunati
- Rješavamo jednadžbu sa vrijednostima iz prethodnog koraka te kao rezultat dobijemo polje brzine



PISO algoritam

- Uvodimo operator $\boldsymbol{H}(\boldsymbol{u})$

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

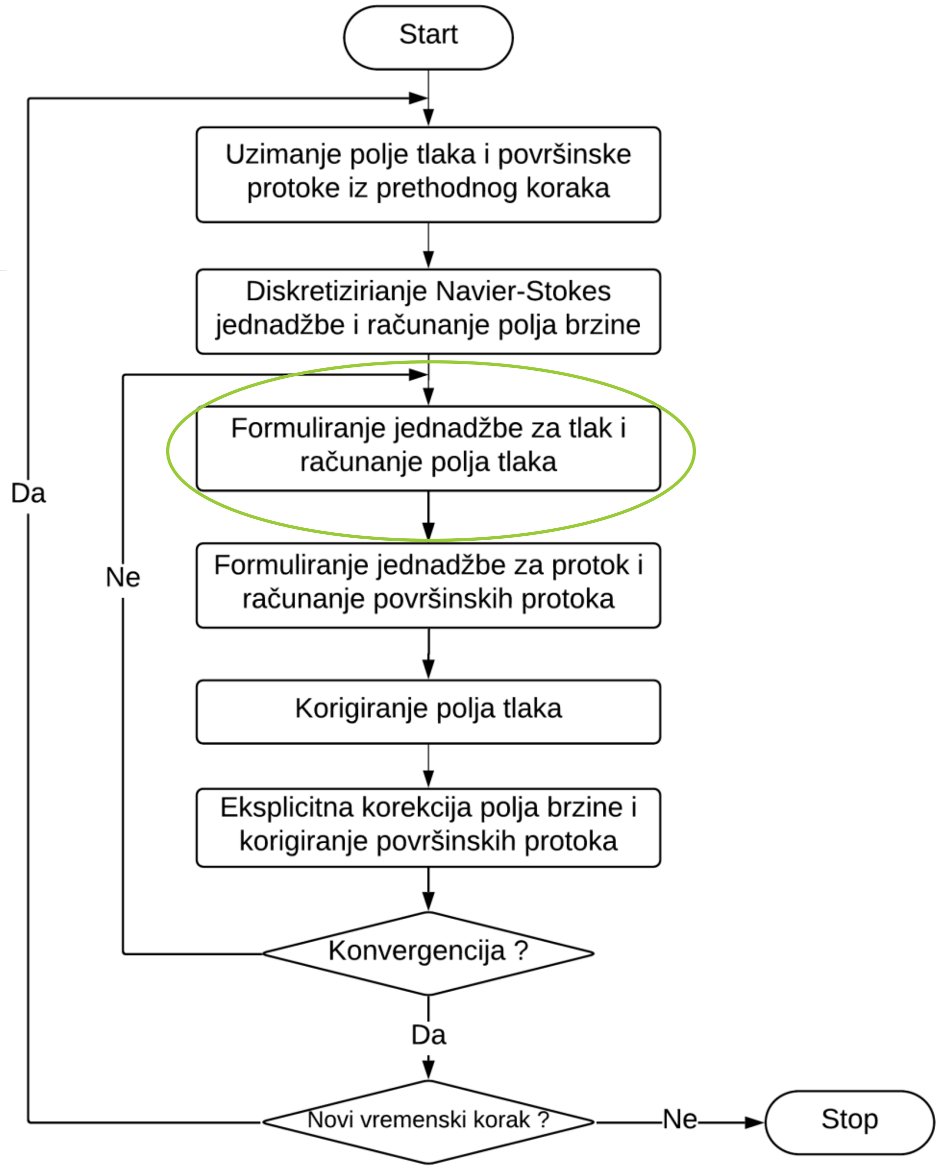
tako da vrijedi

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

- Supstitucijom prethodnog izraza u jednadžbu kontinuiteta dobije se jednadžba za tlak nestlačivog fluida:

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

- Rješavamo jednadžbu tlaka koristeći već izračunato polje brzine



PISO algoritam

- Uvodimo diskretizaciju jednadžbe kontinuiteta s ciljem izvoda jednadžbe za površinski protok F:

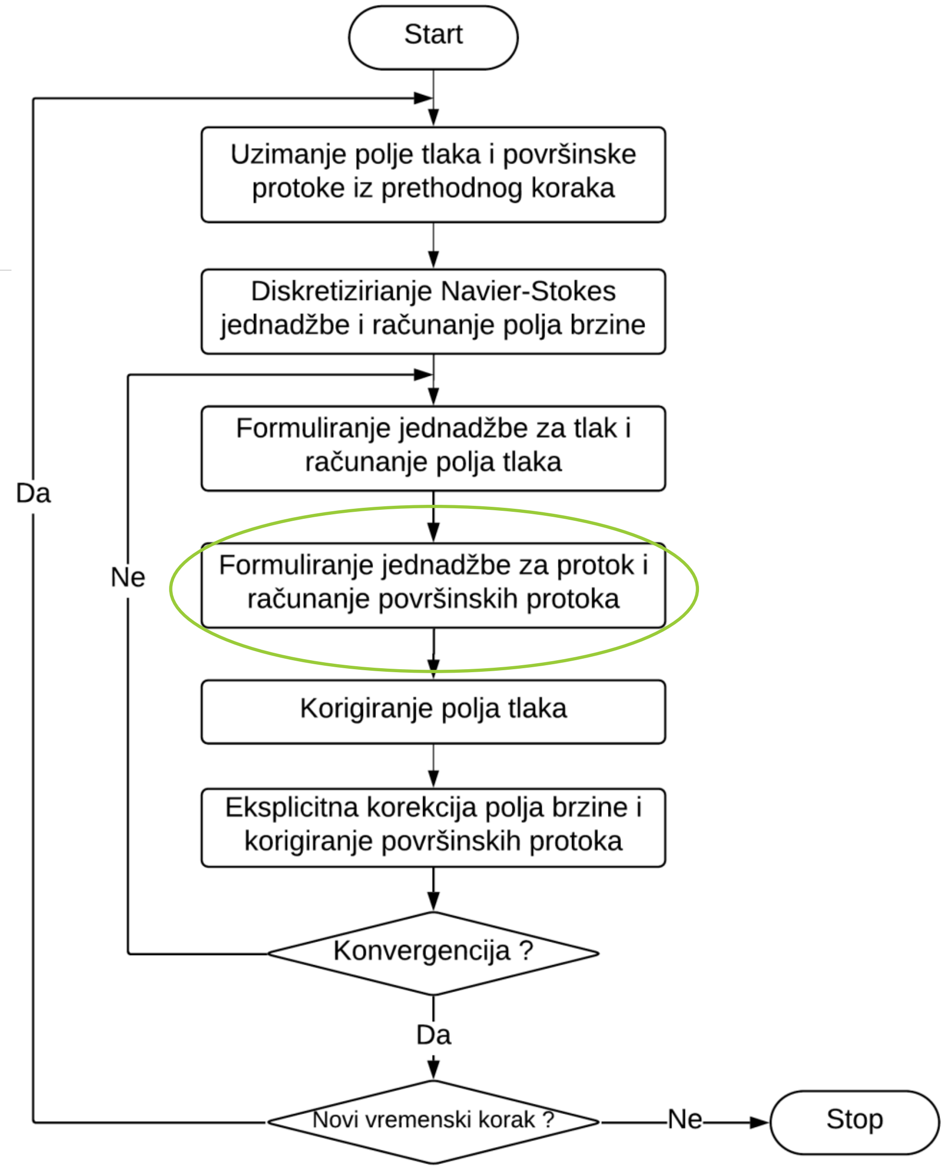
$$\nabla \cdot \boldsymbol{u} = \sum_f F = \sum_f \boldsymbol{s}_f \cdot \boldsymbol{u}$$

gdje $\boldsymbol{s}_f$ ima ulogu vektora smjera površine ćelije

- Supstitucijom izraza za polje brzine u prijašnju jednadžbu dobije se izraz:

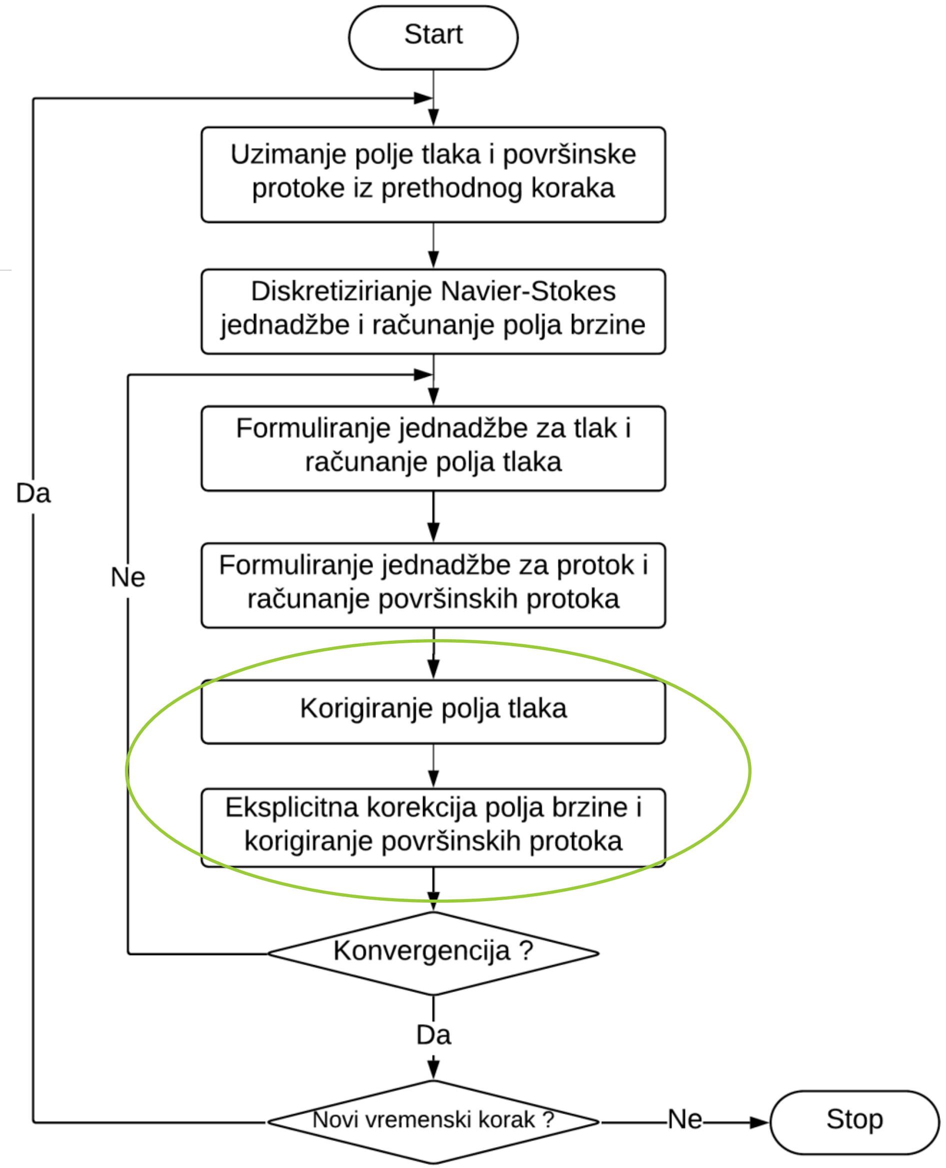
$$F = -\left(a_P^{\boldsymbol{u}}\right)^{-1} \boldsymbol{s}_f \cdot \nabla p + \left(a_P^{\boldsymbol{u}}\right)^{-1} \boldsymbol{s}_f \cdot \boldsymbol{H}(\boldsymbol{u})$$

čijim se rješavanjem računaju površinski protokci



PISO algoritam

- Pomoću izračunatih površinskih protoka rekonstruira se polje brzine te korigira operator $\boldsymbol{H}(\boldsymbol{u})$
- Koristeći dobiveni tlak i korigiranu vrijednost operatora $\boldsymbol{H}(\boldsymbol{u})$ eksplicitno se korigira polje brzine
- Nadalje, izračunaju se novi površinski protoci
- Svi navedeni procesi se ponavljaju do konvergencije



SIMPLE algoritam

- eng. *Semi-Implicit Method for Pressure Linked Equations algorithm*
- Stacionarna metoda namijenjena za korištenje s velikim vremenskim koracima
- Nemamo vremensku derivaciju u jednadžbi zbog velikog vremenskog koraka
- Stoga, potrebno je uvesti podrelaksaciju traženih polja kako bi postigli stabilnost i konvergenciju

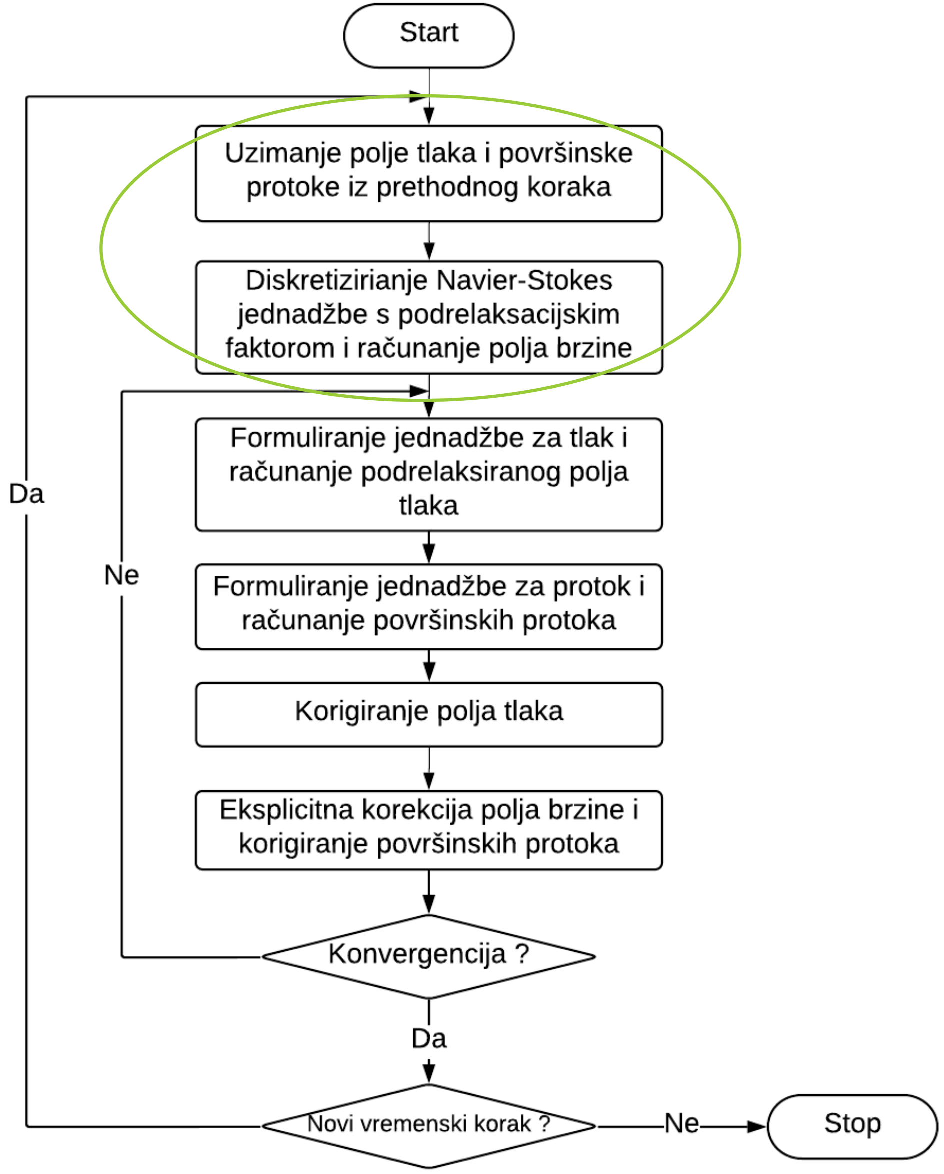
SIMPLE algoritam

- Algoritam počinje preuzimanjem polja tlaka i površinskih protoka iz prethodnog koraka
- Provodimo podrelaksaciju diskretizirane Navier-Stokes jednadžbe:

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

gdje je r_u podrelaksacijski faktor za polje brzine

- Rješavamo jednadžbu sa vrijednostima iz prethodnog koraka te kao rezultat dobijemo polje brzine

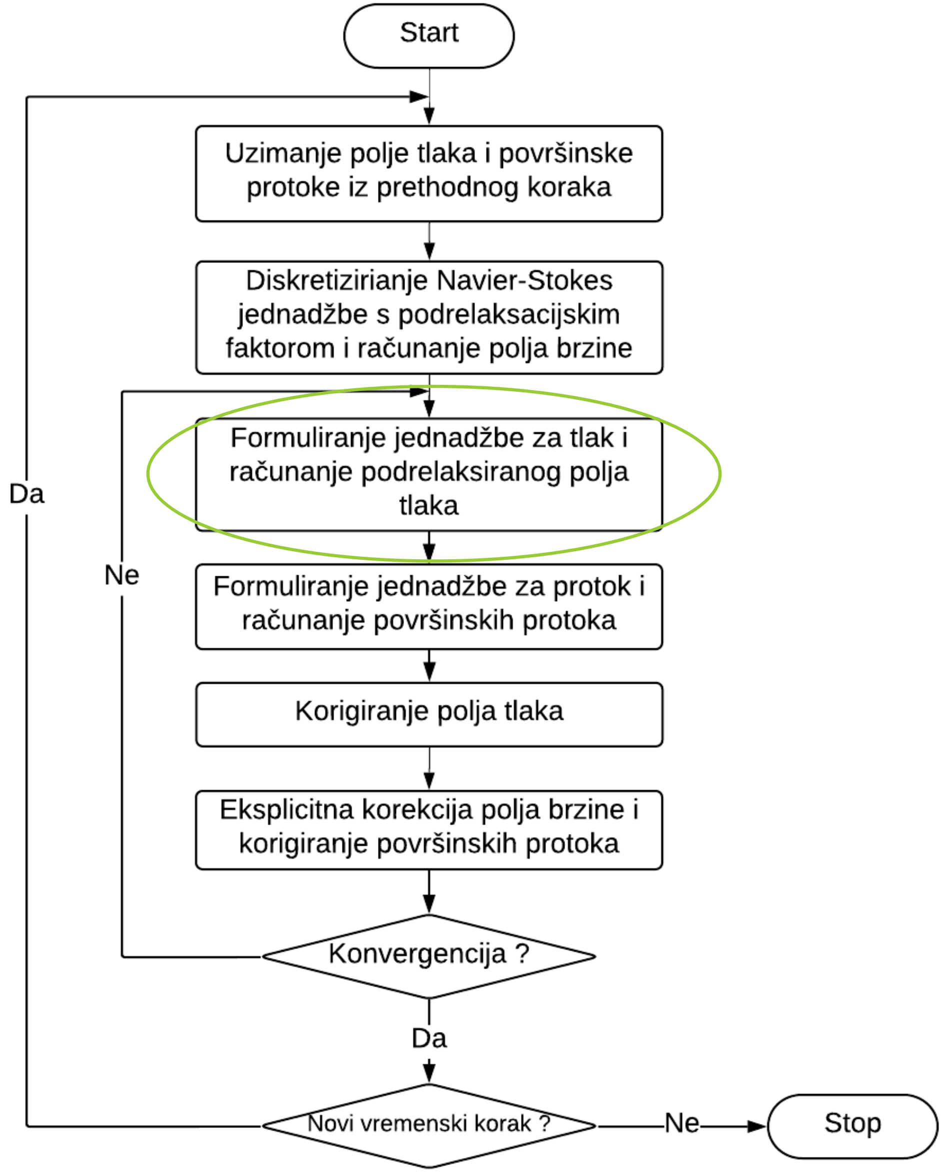


SIMPLE algoritam

- Na isti način kao i kod PISO algoritma, izračuna se operator $\boldsymbol{H}(\boldsymbol{u})$
- Formulira se jednadžba za tlak i izračuna polje tlaka
- Dobiveno polje tlaka podrelaksiramo:

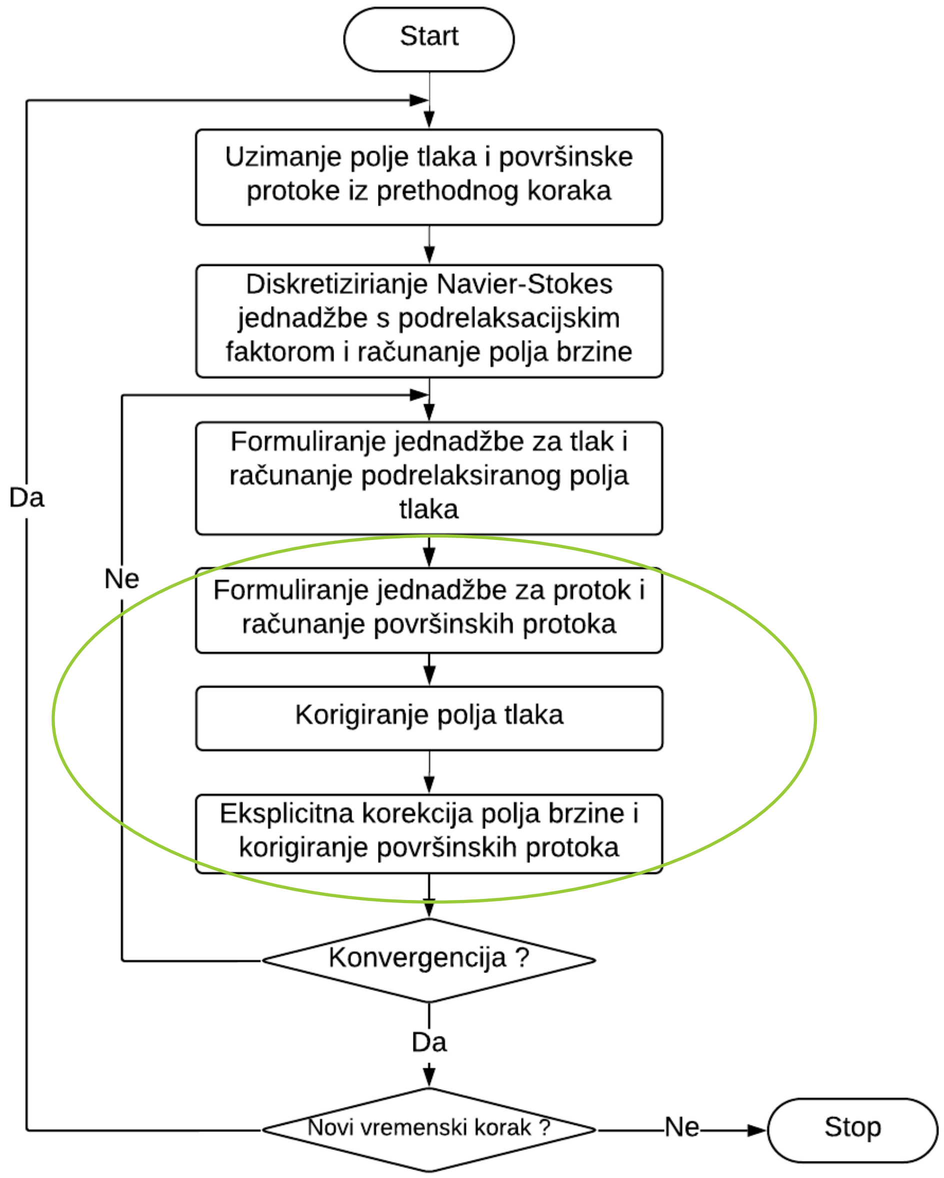
$$p^n = p^o + r_p \cdot (p^p - p^o)$$

  - r_p - podrelaksacijski faktor za polje tlaka
  - p^p- rješenje jednadžbe tlaka
  - p^o - tlak iz prethodnog koraka
  - p^n - podrelaksirani tlak



SIMPLE algoritam

- Kao i kod PISO metode, pomoću dobivenog tlaka računamo površinske protoke
- Korigiramo polje tlaka, polje brzine i površinske protoke
- Svi navedeni procesi se ponavljaju do konvergencije

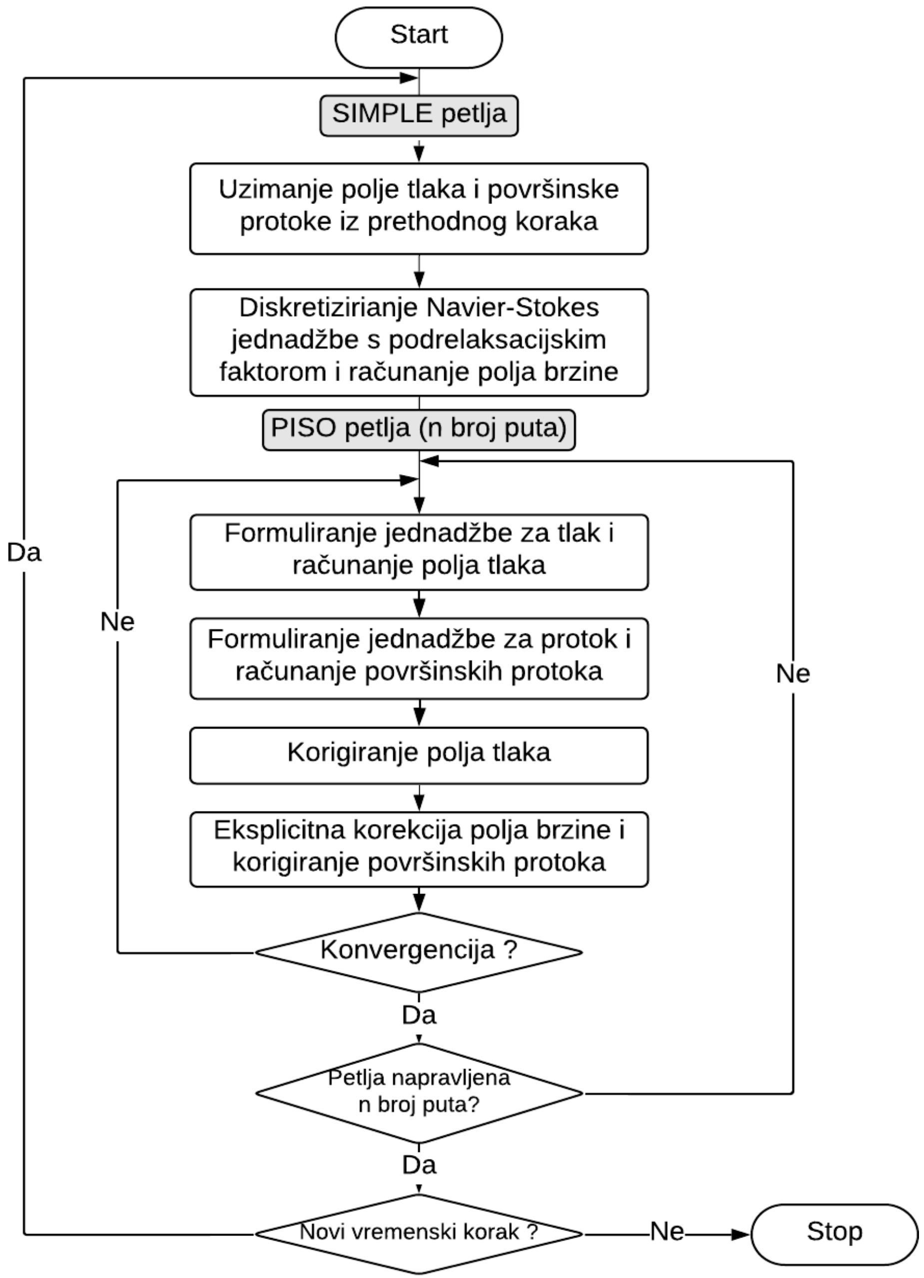


PIMPLE algoritam

- Nastao kao kombinacija PISO i SIMPLE algoritama
- Tranzijenta metoda, ali nakon svakog koraka provodi podrelaksaciju na polju brzine
- Omogućava korištenje većeg vremenskog koraka u odnosu na PISO algoritam
- Posljedica toga je smanjena detaljnost opisa toka

PIMPLE algoritam

- PIMPLE algoritam počinje SIMPLE petljom
- Preuzima se polje tlaka i površinski protoci iz prethodnog koraka
- Diskretizira se N-S jednadžba i izračuna podrelaksirano polje brzine
- Ulazi se u PISO petlju koja se izvršava određeni broj puta, odabrano od strane korisnika u postavkama
- Cijeli postupak se ponavlja do konvergencije



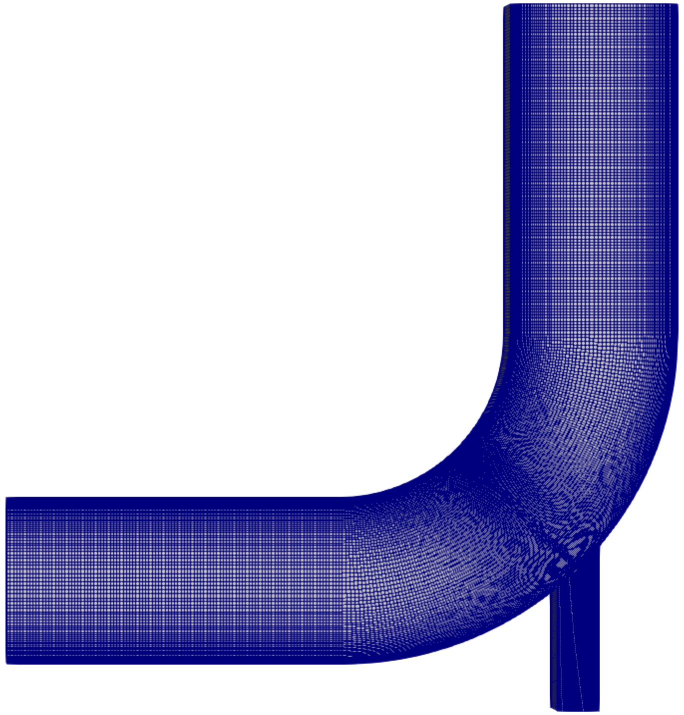
Problem i priprema simulacije

- OpenFOAM je C++ softverski paket otvorenog koda za rješavanje kompleksnih problema u mehanici fluida
- Veliki broj implementiranih numeričkih metoda
- Sadrži alate za generiranje potrebnih mreža kontrolnih volumena
- Za obradu rezultata korišten je ParaView, alat otvorenog koda za vizualizaciju

Problem i priprema simulacije

- Odabrani problem za analizu je prikazana 2D geometrija

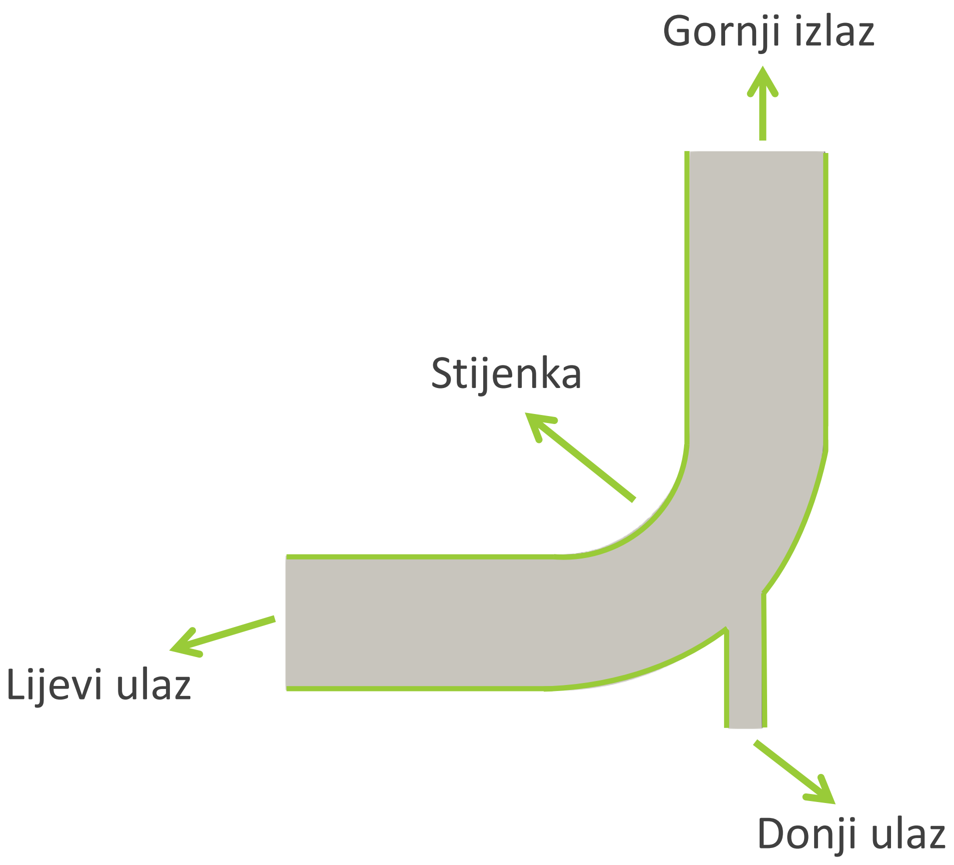
- Generirana mreža na odabranoj geometriji, sadrži otprilike 35 do 40 tisuća ćelija u oba smjera



Problem i priprema simulacije

- Potrebno je postaviti i rubne uvjete:
  - Ulazna granica – određujemo brzinu nastrujavanja fluida
  - Izlazna granica – postavlja se konstanta vrijednost tlaka
  - Zid – nepropusna granica, kod viskoznog strujanja fluid ima brzinu jednaku brzini napredovanja zida

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	$0\,\mathrm{Pa}$
Stijenka	$\boldsymbol{u}$	$0\,\mathrm{m\,s^{-1}}$



Rezultati i analiza

- Karakteristike fluida se definiraju preko kinematičke viskoznosti
- Za naše simulacije odabrana je kinematička viskoznost koja odgovara ulju
- Nakon provedbe simulacije, potrebno je usporediti numeričke metode
- U računalnoj dinamici fluida ne postoji egzaktno rješenje
- Rezultati će biti uspoređeni vizualno i u vidu vremena potrebnog za provedbu same simulacije

Rezultati i analiza

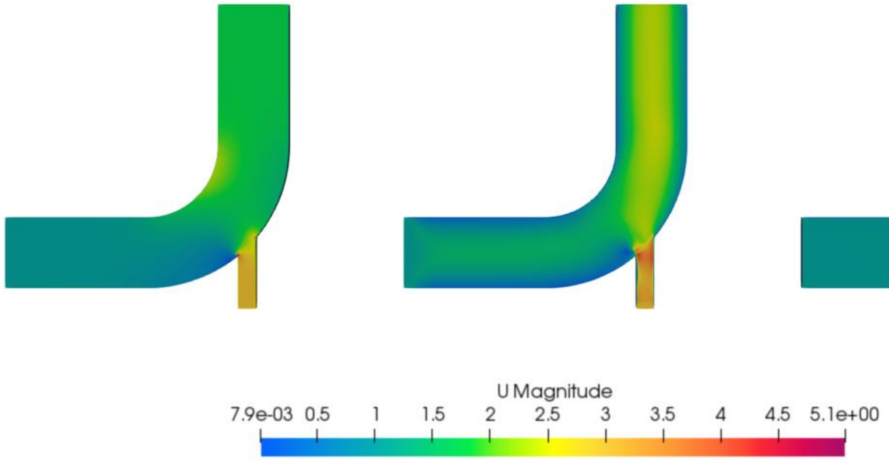
- Očekujemo da je za konvergenciju rezultata PISO metode potrebno najviše vremena, a za konvergenciju rezultata SIMPLE metode najmanje vremena
- Rezultati potvrđuju očekivanja

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

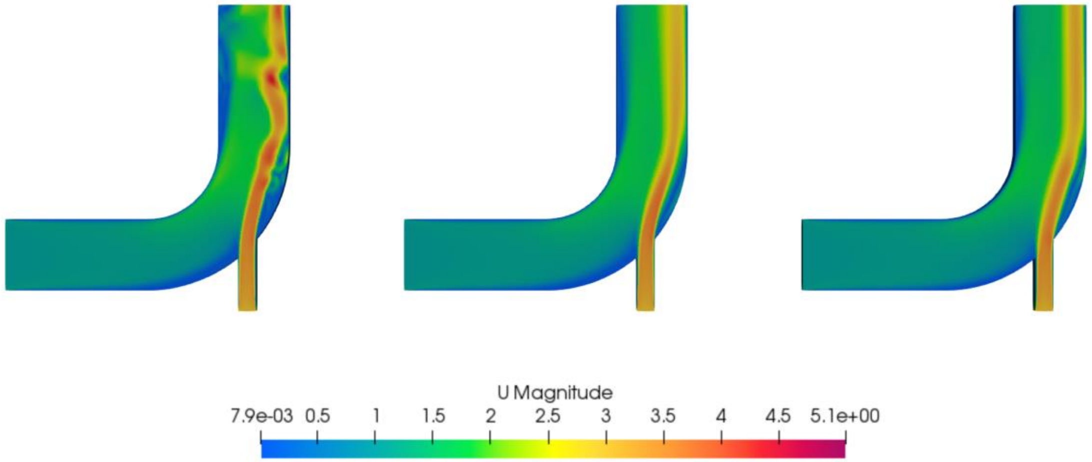
- Iako samo vrijeme izvršavanja ovisi o dostupnoj računalnoj snazi, relativna usporedba je valjana

Rezultati i analiza

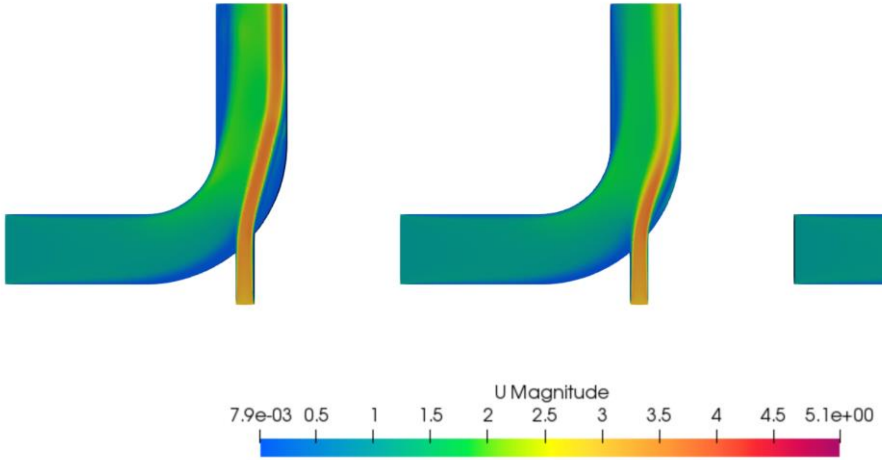
- Rezultate polja brzina će biti prikazani vizualno
- SIMPLE metoda nam nakon nekoliko vremenskih koraka daje konvergirano rješenje
- Za PISO i PIMPLE je potrebno više vremenskih koraka
- Uz početni i zadnji vremenski korak, odabire se približno polovica ukupnog simulacijskog vremena za PISO metodu koja konvergira nakon 220 sekundi



t = 0.1 s

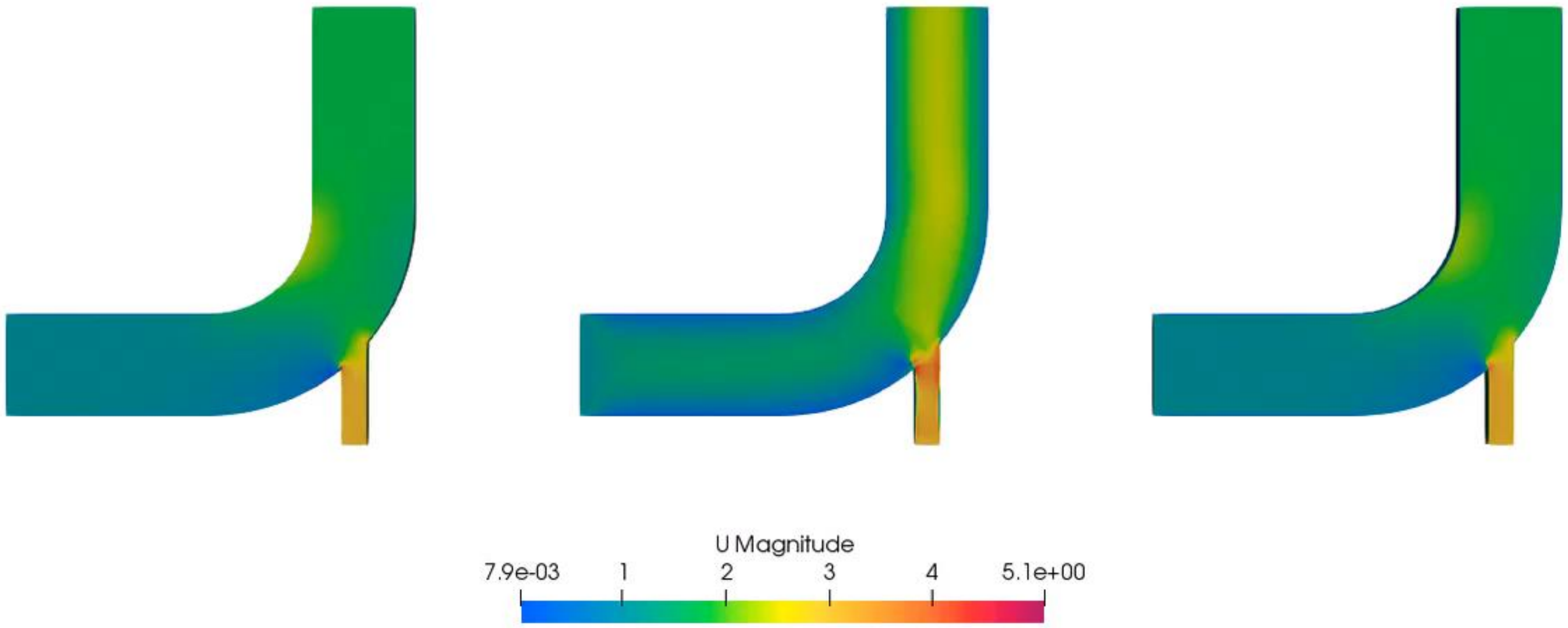


t = 100 s



t = 220 s

- Metode su slijeva nadesno PISO, SIMPLE i PIMPLE algoritam


U Magnitude
7.9e-03
1
2
3
4
5.1e+00

Rezultati i analiza

- Vizualnim pregledom su potvrđene 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
- 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

Zaključak

- Predstavljene su 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
- Simulacije su provedene, rezultati uspoređeni vizualno i u vidu vremena izvršavanja simulacije
- Rezultati su potvrdili očekivanja vezana uz vrijeme izvršavanja simulacije i detaljnost opisa toka

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, Hakan. (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/

Hvala na pažnji!