

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ



Димитрије Рађеновић

МОДЕЛОВАЊЕ КРЕТАЊА ФЛУИДА СА
СЛОБОДНИМ ПОВРШИНАМА
КОРИШЋЕЊЕМ МЕТОДА ЛИЈЕВЕ
АДВЕКЦИЈЕ И ИМПЛИЦИТНИХ
ФУНКЦИЈА

мастер рад

Београд, 2026.

Ментор:

проф. др Весна МАРИНКОВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

проф. др Тијана ШУКИЛОВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

проф. др Игор УЉАРЕВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

Датум одбране:

Наслов мастер рада: Моделовање кретања флуида са слободним површинама коришћењем метода Лијеве адвекције и имплицитних функција

Резиме: Овај мастер рад описује ковекторски приступ за нумеричку симулацију струјања флуида са слободним површинама, са посебним освртом на очување вртложних структура током временске еволуције. Традиционалне методе третирају брзину као векторско поље, чија адвекција заснована на материјалном изводу често уноси нумеричку дифузију. Ово нарушава Келвинову теорему о очувању циркулације, што у дуготрајним симулацијама неизбежно доводи до деградације вртлога. Губитак енергије на малим скалама представља кључни изазов у рачунарској графици, јер директно умањује визуелну уверљивост симулација флуида. Ковекторски формализам решава наведени проблем формулишући брзину флуида као диференцијалну 1-форму, која се адвектује применом Лијевог извода, чиме се елиминишу главни извори нумеричке дисипације вртложности.

У оквиру овог мастер рада развијен је тродимензионални рачунарски код који, према сазнањима аутора, представља прву имплементацију ковекторског формализма која користи Ојлеров приступ за симулацију струјања флуида са слободним површинама. За праћење граничне површине која раздваја различите фазе, коришћена је метода ниво скупа заснована на имплицитним функцијама, која обезбеђује робусно моделовање њене временске еволуције. Развијени код је верификован на тест случајевима, укључујући Тејлор–Гринове и торусне вртлоге, слободан пад водене сфере у базен, као и струјање флуида, који истиче из извора смештених на суседним вертикалним зидовима. Добијени резултати потврђују да ковекторски приступ омогућава стабилне и физички конзистентне симулације струјања флуида, чиме се постиже значајан напредак у њиховом нумеричком моделовању са директном применом у рачунарској графици и визуелизацији.

Кључне речи: нумеричка симулација, слободна површина, метода ниво скупа, ковекторски приступ, Лијева адвекција, имплицитне функције.

Садржај

1	Увод	1
2	Основе динамике флуида	3
2.1	Основни појмови	4
2.2	Једначине кретања флуида	5
2.3	Вртложност и вртложна формулација једначина кретања флуида	8
2.4	Гранични услови	10
3	Нумеричке методе за решавање једначина кретања флуида	13
3.1	Лагранжов и Ојлеров приступ решавању једначина кретања флуида	13
3.2	Дискретизација флуидних поља	15
3.3	Метода раздвајања оператора	20
3.4	Семи-Лагранжова адвекција	23
3.5	Пројекција притиска	26
3.6	Стабилност прорачуна	29
4	Двофазни ток флуида	31
4.1	Дефиниција и класификација граничних површина између фаза	31
4.2	Гранична површина у двофазним токовима	32
4.3	Методе засноване на имплицитним функцијама	35
4.4	Флуиди са слободном површином	41
5	Ковекторски приступ анализи кретања флуида	43
5.1	Историјски корени методе	43
5.2	Основни појмови теорије диференцијалних форми	44
5.3	Лијев извод и Лијева адвекција	53

6	Имплементација флуидног симулатора	59
6.1	Прорачунске мреже	60
6.2	Решавачи	62
7	Резултати симулација	70
7.1	Тејлор–Гринов вртлог	70
7.2	Торусни вртлог	73
7.3	Пад водене сфере у базен	77
7.4	Извори флуида на суседним вертикалним зидовима	79
8	Закључак	81
	Библиографија	83

Глава 1

Увод

Симулација динамике флуида представља један од најизазовнијих проблема у рачунарској графици и научној визуелизацији [1]–[4]. Током протеклих деценија, развој нумеричких метода за решавање *Навије–Стоксових* (енг. *Navier–Stokes*) једначина омогућио је генерисање визуелно уверљивих анимација течности, дима и експлозија, које су данас стандард у филмској индустрији и видео-играма [5, 6]. Ипак, проналажење баланса између нумеричке стабилности, рачунске ефикасности и физичке тачности остаје отворен математички проблем. Посебно је критично очување кључних физичких инваријанти, попут вртложности и циркулације, током дуготрајних симулација.

У домену Ојлерових метода, које користе фиксну прорачунску мрежу, доминантан приступ за дискретизацију конвективног члана је *семи-Лагранжова* (енг. *semi-Lagrangian*) адвекција [7]. Иако ова метода гарантује стабилност и омогућава велике временске кораке, она уноси значајну нумеричку дифузију. Ово доводи до постепеног слабљења вртложних структура и нарушавања *Келвинове* (енг. *Kelvin*) теореме о очувању циркулације за идеалне флуиде [8], што резултира физички неконзистентним и визуелно сиромашним симулацијама неадекватним за примене у рачунарској графици. Уверљивост специјалних ефеката у филмовима и видео-играма директно зависи од очувања вртложних структура, па развој геометријски конзистентних метода представља кључни захтев савремене VFX (енг. *Visual Effects*) продукције [9, 10].

Као одговор на поменута ограничења, у овом мастер раду представљен је *ковекторски* *иприслиуи* моделовању нестишљивих и невискозних флуида [11]. Уместо класичног третирања брзине као векторског поља, брзина се формулише као диференцијална 1-форма. Адвекциони корак се затим реализује

применом Лијевог извода, чиме се геометријска структура континуалних једначина имплицитно преноси на дискретни ниво. Овај формализам структурно осигурава очување циркулације и елиминише главне изворе нумеричке дисипације, омогућавајући стабилну еволуцију вртложних структура чак и након великог броја временских корака.

Иако је ковекторски формализам оригинално развијен за једнофазне флуиде без слободне површине [11], главни допринос овог мастер рада лежи у његовом проширењу и примени на флуиде са слободним површинама. У ту сврху, за праћење међуповршине фазног раздвајања примењена је *метода ниво скупа* (енг. *Level Set Method*) заснована на имплицитним функцијама [12, 13]. Да би се одржала метричка прецизност функције означеног растојања, имплементирана је периодична реиницијализација коришћењем *метода брзог марширања* (енг. *Fast Marching Method*) [12].

Нумеричка имплементација ковекторског приступа реализована је развојем тродимензионалног кода заснованог на *помереној* (енг. *staggered*) *МАС* (енг. *marker-and-cell*) мрежи. Адвекциони корак је модификован ковекторском корекцијом која укључује транспоновани Јакобијан инверзног пресликавања флуидног тока, док је пројекција притиска решавана *методом конјугованих градијената* ради задовољења услова нестишљивости. Код је оптимизован коришћењем OpenMP паралелизације, што осигурава скалабилност на вишејезгарним системима. Верификација на тест случајевима вртлога, *Тејлор–Гриновом* (енг. *Taylor–Green vortex*) и *торусном* (енг. *torus vortex*), потврдила је ефикасност ковекторске шеме у очувању вртложности.

Мастер рад је организован у осам поглавља. Након увода, у другом поглављу изложени су теоријски основи динамике флуида, укључујући Навије–Стоксове једначине, дефиницију вртложности и Келвинову теорему. У трећем поглављу описане су нумеричке методе за дискретизацију једначина кретања, док се у четвртном разматрају моделовање двофазних токова и метод ниво скупа за праћење слободних површина. Пето поглавље је посвећено ковекторском приступу, теорији диференцијалних форми и Лијевој адвекцији, на којима се заснива овај рад. Шесто поглавље описује архитектуру развијеног тродимензионалног кода за симулацију нестишљивих и невискозних флуида са слободним површинама, док су у седмом приказани и анализирани резултати извршених симулација. На крају, закључак даје осврт на целокупан рад и указује на могуће правце даљег истраживања.

Глава 2

Основе динамике флуида

Материја у природи има дискретну микроструктуру, састављену од атома и молекула раздвојених простором, чије се понашање може описати математичким моделима заснованим на законима физике. Симулација флуида на микроскопском нивоу, ради моделовања макроскопских појава, захтева супер-рачунаре и прорачуне који могу трајати недељама или месецима. Рачунарска графика тежи компромису између ефикасности и прецизности моделовања, па се користи механика флуида утемељена на хипотези континуума.

Према хипотези континуума, флуид се моделује као непрекидна средина без празнина, чиме се занемарује молекуларна структура, а физичке величине попут густине, брзине и притиска постају непрекидне функције простора и времена. У овом приступу, *честица флуида* (енг. *fluid parcel*) представља елемент запремине довољно велик да садржи статистички репрезентативан број молекула, а довољно мали да се може третирати као тачка у којој су све физичке величине поља локално дефинисане.

Хипотеза континуума, у комбинацији са законима очувања масе, импулса и енергије, представља теоријску основу за извођење основних једначина динамике флуида, почев од Ојлерових до Навије-Стоксових [7]. У механици флуида развијени су различити модели који се заснивају на доминантним карактеристикама струјања, како би се омогућило поједностављење или детаљније разматрање проблема [14]:

- Нестишљив флуид — модел у коме се густина сматра константном у простору и времену. Примењује се код течности и споро крећућих гасова (где су промене притиска мале);
- Идеалан (савршен) флуид — модел невискозног флуида без унутрашњег

трења. Омогућава аналитичка решења основних једначина динамике флуида (нпр. Ојлерове једначине) за ограничен скуп поједностављених проблема, али не узима у обзир дисипацију енергије;

- Стишљив флуид — модел у коме се густина мења у зависности од притиска и температуре и где доминирају еластичне силе. Кључан је у аеродинамици, при струјању великим брзинама и у динамици гасова;
- Реалан флуид — најкомплетнији модел који узима у обзир вискозност и стишљивост, али је математички најзахтевнији, па се решења добијају искључиво нумерички.

Наведени модели представљају хијерархијски оквир у механици флуида, при чему се избор одговарајућег модела заснива на карактеристикама протока, брзини, притиску, температури и жељеној тачности анализе.

2.1 Основни појмови

У механици флуида разликују се две основне категорије сила:

- Запреминске (масене) силе — делују на сваку честицу флуида унутар целе запремине. Пропорционалне су маси и у једначинама кретања јављају се као извор убрзања;
- Површинске силе — делују на границама између делова флуида, или између флуида и спољашњих површина (нпр. зидова цеви). Обухватају силе притиска и вискозног трења, а њихова расподела се описује тензором напона.

Густина флуида ρ се дефинише као маса по јединици запремине. У теорији струјања, она је фундаментални параметар у законима очувања масе и импулса. За стишљиве флуиде, густина је функција притиска и температуре, па је за потпуно дефинисање термодинамичког система неопходна једначина стања. Насупрот томе, код нестишљивих флуида густина је константна, па је поље брзине недивергентно, односно дивергенција вектора брзине је једнака нули у свакој тачки, што знатно поједностављује опис струјања.

Динамичка вискозност μ је фундаментално својство флуида које одређује отпор релативном кретању суседних слојева услед унутрашњег трења. У зависности од тога да ли се њена вредност мења при струјању, флуиди се деле на њутновске и нењутновске. Код њутновских флуида (нпр. вода, ваздух и индустријска уља при стандардним условима) вискозност је константна и

зависи искључиво од притиска и температуре, што поједностављује њихов математички опис. Код нењутновских флуида (нпр. крв, полимерни раствори и пасте) вискозност зависи од услова струјања, што захтева примену сложенијих модела.

Кинематичка вискозност ν се дефинише као количник динамичке вискозности и густине флуида. За одређивање режима струјања користи се Рејнолдсов број, који представља однос инерцијалних и вискозних сила и обрнуто је сразмеран кинематичкој вискозности. За мале вредности Рејнолдсовог броја, струјање је ламинарно и одликује се слојевитом структуром са паралелним струјницама. Са повећањем Рејнолдсовог броја, развој хидродинамичких нестабилности иницира прелаз у турбулентни режим кога карактерише хаотична, вртложна структура.

Притисак p се дефинише као интензитет нормалне компоненте силе по јединици површине. У флуидима настаје услед молекуларних судара и интеракција честица. У стању мировања флуид не трпи тангенцијалне напоне, па је притисак изотропан, односно једнак у свим правцима у датој тачки. При струјању, укупни тензор напона чине изотропни притисак и тангенцијални (вискозни) напони сразмерни градијентима брзине, који условљавају дисипацију енергије и развој турбуленције.

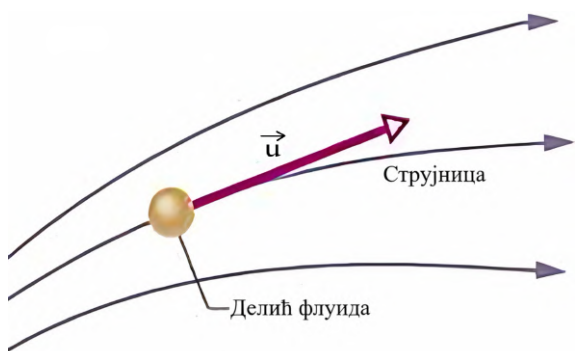
2.2 Једначине кретања флуида

У овом делу представимо основне једначине динамике флуида, Ојлерове и Навије–Стоксове, које описују кретање флуида и чине математички оквир за анализу тока. Посебно су погодне за приказ једноставнијих, ламинарних (у паралелним слојевима) режима струјања. Ипак, до данас, није доказано да ли тродимензионална решења Навије–Стоксових једначина увек постоје, па је овај проблем уврштен међу миленијумске проблеме Клејовог математичког института [15].

Решавањем одговарајућег система једначина механике флуида добија се векторско поље брзина, које се користи за визуелизацију струјања, или даљу математичку анализу. Постоје два основна приступа за описивање кретања флуида: Лагранжов, из кога потиче појам трајекторије, и Ојлеров, са којим је повезан појам струјнице.

Лагранжова метода заснива се на праћењу честица флуида и вредности

величина које их карактеришу. Трајекторија, или путања честице флуида, представља криву у простору коју честица описује од почетног до произвољног тренутка. Главни недостатак ове методе је потреба да се дефинишу једначине кретања за бесконачно много честица у пољу струјања.



Слика 2.1: Приказ струјнице: у датом тренутку, вектор брзине делића флуида тангентан је на струјницу [16].

У Ојлеровој методи посматрају се промене физичких величина у фиксним тачкама поља струјања. Кроз сваку тачку поља струјања током времена пролазе различити делићи флуида, носећи своје физичке карактеристике. Струјница је векторска линија поља брзине (Слика 2.1), коју у одређеном тренутку чине различити делићи флуида чији су вектори брзине тангентни на ту линију [16]. Насупрот томе, трајекторија представља стварну путању појединачног делића флуида.

Стационарно струјање је оно код кога се брзина и остале физичке величине у свакој фиксној тачки простора не мењају током времена, без обзира на то који делић флуида у том тренутку пролази кроз њу. При стационарном струјању струјнице и трајекторије се поклапају. Код нестационарног струјања струјнице се мењају у времену и не поклапају се са трајекторијама. На пример, ако ветар мења смер док дим излази из димњака, струјница показује само тренутни смер ветра, док трајекторија представља стварни (искривљени) пут честице дима настао под утицајем претходних смерова ветра.

Приликом извођења основних једначина механике флуида користе се појмови *материјалне* и *контролне запремине*. Из Лагранжове методе произилази појам материјалне запремине V_m , односно издвојене запремине флуида која садржи увек исте делиће флуида и ограничена је материјалном површином A_m . Појам контролне запремине, који потиче из Ојлерове методе, односи се на фиксну запремину V , ограничену фиксном површином A .

Навије–Стоксове и Ојлерове једначине

Навије–Стоксове једначине представљају систем нелинеарних парцијалних диференцијалних једначина другог реда које описују кретање вискозних флуида. У општем случају решавају се апроксимативним нумеричким методама, док су аналитичка решења могућа искључиво у веома поједностављеним, идеализованим ситуацијама. Са друге стране, Ојлерове једначине, које описују кретање невискозних флуида, чине систем парцијалних диференцијалних једначина првог реда. Оне се заснивају на законима очувања масе, импулса и енергије, који су, према теореме *Еми Неџер* (енг. *Emmy Noether*), повезани са симетријама система: очување енергије са хомогеношћу времена, а очување импулса са хомогеношћу простора [17].

Нестипљиви токови хомогених флуида у целом простору $\mathbb{R}^n$, $n = 2, 3$, представљају решења система једначина [7]:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u} + \mathbf{f}, \quad (2.1)$$

$$\operatorname{div} \mathbf{u} = \nabla \cdot \mathbf{u} = 0, \quad (\mathbf{x}, t) \in \mathbb{R}^n \times [0, \infty), \quad (2.2)$$

$$\mathbf{u}|_{t=0} = \mathbf{u}_0, \quad \mathbf{x} \in \mathbb{R}^n, \quad (2.3)$$

где су: $\mathbf{u}(\mathbf{x}, t) \equiv (u^1, u^2, \dots, u^n)^T$ вектор брзине флуида, $p(\mathbf{x}, t)$ притисак, ρ густина флуида, ν кинематичка вискозност, $\operatorname{div} \mathbf{u}$ дивергенција векторског поља $\mathbf{u}$, ∇ оператор градијента и Δ Лапласов оператор. Члан који описује убрзање које потиче од спољашњих сила, нпр. гравитације, означен је са $\mathbf{f}$. Напоменимо да се у рачунарској графици компоненте вектора брзине обично означавају са (u, v, w) .

Конвективни или материјални извод D/Dt , представља извод дуж трајекторије честице:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \sum_{j=1}^n u^j \frac{\partial}{\partial x^j}. \quad (2.4)$$

Тада се лева страна једначине (2.1) може написати у облику [8]:

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{\partial\mathbf{u}}{\partial t} + \underbrace{\begin{bmatrix} \frac{\partial u^1}{\partial x^1} & \frac{\partial u^1}{\partial x^2} & \frac{\partial u^1}{\partial x^3} \\ \frac{\partial u^2}{\partial x^1} & \frac{\partial u^2}{\partial x^2} & \frac{\partial u^2}{\partial x^3} \\ \frac{\partial u^3}{\partial x^1} & \frac{\partial u^3}{\partial x^2} & \frac{\partial u^3}{\partial x^3} \end{bmatrix}}_{\text{Јакобијева матрица } \nabla \mathbf{u}} \mathbf{u}. \quad (2.5)$$

Треба нагласити да се уместо ознака $\mathbf{u}(\mathbf{x}, t) \equiv (u^1, u^2, u^3)^T$ за вектор брзине флуида и (x^1, x^2, x^3) за координате, у рачунарској графици најчешће користе $\mathbf{u}(\mathbf{x}, t) \equiv (u, v, w)^T$ и (x, y, z) , које ћемо користити у даљем тексту. Када је вредност кинематичке вискозности ν већа од нуле, једначине (2.1) су Навије-Стоксове, док се за $\nu = 0$ свде на Ојлерове једначине. Једначина (2.2) изражава услов нестишљивости флуида, при чему је стање таквог флуида дефинисано помоћу четири променљиве: три компоненте вектора брзине и притиска. Почетни проблем задат једначинама (2.1)–(2.3) садржи временске изводе само за три од укупно четири непознате функције.

2.3 Вртложност и вртложна формулација једначина кретања флуида

Вртложно кретање флуида

Вртложност има кључну улогу у динамици флуида, па се струјање често анализира кроз еволуцију њеног поља. Поље вртложности $\boldsymbol{\omega}$ је повезано са пољем брзине $\mathbf{u}$ на следећи начин [7]:

$$\boldsymbol{\omega} = \text{rot } \mathbf{u} = \nabla \times \mathbf{u}, \quad (2.6)$$

одакле следи да вртложност за недивергентно поље ($\nabla \cdot \mathbf{u} = 0$) задовољава:

$$\nabla \cdot \boldsymbol{\omega} = 0. \quad (2.7)$$

У Декартовом координатном систему релација (2.6) даје следеће везе између компоненти брзине (u, v, w) и компоненти вртложности $(\omega_x, \omega_y, \omega_z)$:

$$\omega_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \quad \omega_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \quad \omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \quad (2.8)$$

У дводимензионалном случају, поље вртложности има само једну ненулту компоненту ω_z ортогоналну на (x, y) раван, чиме се аутоматски задовољава услов недивергентности (2.7).

Циркулација Γ вртложног поља око затворене криве L , која окружује површину S чији је вектор нормале $\mathbf{n}$, дефинисана је као линијски интеграл:

$$\Gamma = \int_L \mathbf{u} \cdot d\mathbf{r} = \int_S \boldsymbol{\omega} \cdot \mathbf{n} dS. \quad (2.9)$$

Према Келвиновој теореме, циркулација брзине дуж затворене контуре која се креће са флуидом не мења се током времена [7]:

$$\frac{D\Gamma}{Dt} = 0, \quad (2.10)$$

уколико су испуњени следећи услови:

- невискозност флуида, јер вискозност представља главни механизам стварања и дисипације вртлога,
- конзервативност спољашњих сила (нпр. гравитационе силе),
- баротропност флуида, тј. густина је једнозначна функција притиска,
- глаткост тока флуида, односно одсуство прекида попут ударних таласа.

То значи да се вртлози у таквом флуиду не могу спонтано појавити нити ишчезнути, већ могу постојати само ако су задати почетним или уведени граничним условима.

Овде ће укратко бити приказан једноставан опис нестишљивог вртложног струјања флуида. Свако глатко поље брзине $\mathbf{u}(\mathbf{x}, t)$ има Тејлоров развој око фиксне тачке $(\mathbf{x}_0, t_0)$:

$$\mathbf{u}(\mathbf{x}_0 + \mathbf{h}, t_0) = \mathbf{u}(\mathbf{x}_0, t_0) + \nabla \mathbf{u}(\mathbf{x}_0, t_0) \mathbf{h} + \mathcal{O}(|\mathbf{h}|^2), \quad \mathbf{h} \in \mathbb{R}^3. \quad (2.11)$$

Јакобијева матрица $\nabla \mathbf{u}$ има симетрични део $\mathcal{D}$ и антисиметрични део Ω :

$$\mathcal{D} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T), \quad (2.12)$$

$$\Omega = \frac{1}{2} (\nabla \mathbf{u} - \nabla \mathbf{u}^T). \quad (2.13)$$

$\mathcal{D}$ се назива матрица инфинитезималне стопе деформације, док се Ω назива матрица инфинитезималне стопе ротације. Ако је струјање нестишљиво ($\operatorname{div} \mathbf{u} = 0$), онда је траг $\operatorname{tr} \mathcal{D} = \sum_i d_{ii} = 0$. Може се показати да за вртложност $\boldsymbol{\omega}$ важи [7]:

$$\Omega \mathbf{h} = \frac{1}{2} \boldsymbol{\omega} \times \mathbf{h}, \quad \forall \mathbf{h} \in \mathbb{R}^3. \quad (2.14)$$

Користећи Тејлоров развој (2.11) и релацију (2.14), добија се:

$$\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}(\mathbf{x}_0, t_0) + \frac{1}{2} \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0) + \mathcal{D}(\mathbf{x} - \mathbf{x}_0) + \mathcal{O}(|\mathbf{x} - \mathbf{x}_0|^2), \quad (2.15)$$

где је $\mathcal{D}$ (симетрична) матрица деформације чији је траг $\operatorname{tr} \mathcal{D} = 0$. Чланови на десној страни релације (2.15) имају физичку интерпретацију: први одговара транслацији, други инфинитезималној ротацији, а трећи инфинитезималној деформацији. Према томе, свако нестишљиво вртложно поље брзине $\mathbf{u}(\mathbf{x}, t)$ може се представити као сума брзина које одговарају бесконачно малој транслацији, ротацији и деформацији.

Вртложна формулација једначина кретања флуида

Применом оператора ротор ($\nabla \times$) на Навије-Стоксове једначине (2.1), уз (2.6) и релацију $\nabla \times [(\mathbf{u} \cdot \nabla)\mathbf{u}] = (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla)\mathbf{u}$, која важи за нестишљиве флуиде, добија се динамичка једначина за вртложност $\boldsymbol{\omega}$ у којој не фигурише притисак (деталји извођења се могу наћи у [7]):

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \Delta \boldsymbol{\omega} + \nabla \times \mathbf{f}. \quad (2.16)$$

Уколико је флуид нестишљив и невискозан, ова једначина добија знатно једноставнији облик:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nabla \times \mathbf{f}. \quad (2.17)$$

Решавањем једначине (2.17), уз одговарајуће почетне и граничне услове, одређује се вртложност флуидног кретања. Међутим, то обично није довољно, јер се захтева и вектор брзине, а понекад и притисак. Реконструкција вектора брзине $\mathbf{u}$ на основу вртложности $\boldsymbol{\omega}$ је сложен и нумерички захтеван процес, којим се овај рад не бави. Ковекторска метода, која је главна тема овог рада, елиминира потребу за тим.

2.4 Гранични услови

Већина изазова у нумеричкој симулацији струјања флуида лежи у адекватном задавању граничних услова. У наставку ћемо се фокусирати на два основна типа граничних услова: *чврсте зидове* (енг. *solid walls*) и *слободне површине* (енг. *free surfaces*) [2].

Гранични услов чврстог зида јавља се тамо где се флуид додирује са чврстим телом. Најједноставније га је формулисати преко брзине: флуид не сме продирати у чврсто тело нити излазити из њега кроз контактну површину, па мора важити да је нормална компонента брзине флуида једнака нули:

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad (2.18)$$

где је $\mathbf{u}$ вектор брзине флуида, а $\mathbf{n}$ јединични вектор нормале на површину зида, усмерен ка флуиду. Ситуација је сложенија ако се тело креће. У општем случају, нормална компонента брзине флуида мора да буде једнака нормалној компоненти брзине чврстог тела, тако да релативна брзина између флуида и

тела има нулту нормалну компоненту [2]:

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{u}_{\text{solid}} \cdot \mathbf{n}, \quad (2.19)$$

где је $\mathbf{u}_{\text{solid}}$ брзина чврстог тела на граници. Услов (2.19), познат као *услов без прилепљивања* (енг. *no-stick condition*), ограничава искључиво нормалну компоненту брзине, док се флуид слободно креће дуж граничне површине у правцу тангенте. Другим речима, тангенцијална компонента брзине флуида може бити независна од тангенцијалне компоненте брзине чврстог тела.

Што се тиче притиска на чврстом зиду, његова вредност није слободно задата, већ је одређена тако да се обезбеди нестишљивост флуида и да се испуне гранични услови на чврстом зиду. У једначини кретања (2.1), члан $\nabla p / \rho$ важи и на граници. Како би притисак могао да контролише нормалну компоненту брзине $\mathbf{u} \cdot \mathbf{n}$ на чврстом зиду, мора бити задовољен одговарајући услов за скаларни производ $\nabla p \cdot \mathbf{n}$, познат и као нормални извод притиска.

Овим је дефинисан гранични услов на чврстом зиду за невискозан флуид. Међутим, у присуству вискозности, ситуација постаје сложенија. Вискозни ефекти узрокују да се флуид „прилепи“ за чврсту површину, што значи да се и тангенцијална компонента брзине мора ускладити са брзином зида. Ово се назива *условом без клизања* (енг. *no-slip condition*), према коме вектор брзине флуида на граници мора да се поклапа са вектором брзине чврстог тела: $\mathbf{u} = \mathbf{0}$ ако је зид непокретан, или $\mathbf{u} = \mathbf{u}_{\text{solid}}$ ако се чврсто тело креће.

У пракси, утицај вискозности и ефекти услова прилепљивања ограничени су на изузетно танак слој уз зид, такозвани гранични слој. Ван тог слоја, вискозни чланови постају занемарљиви, па се флуид понаша приближно као невискозан. У том региону, динамика флуида је добро апроксимирана коришћењем *услова слободног клизања* (енг. *free-slip condition*), при коме је дозвољено тангенцијално кретање флуида дуж зида, док је нормална компонента брзине једнака нули.

Услови на површинама кроз које флуид улази у систем и излази из система су кључни за правилно моделовање тока. На улазу, најчешће се користе *услови константне брзине* (енг. *velocity inlet*) или *сталног протока* (енг. *mass flow inlet*). На излазу, доминира *услов константног притиска* (енг. *pressure outlet*), којим се задаје референтни притисак (нпр. атмосферски), или пад притиска кроз систем. Да би се избегла нестабилност услед рефлексије таласа, примењују се *услови слободног протока* (енг. *outflow*). У сложенијим

системима, попут оних са слободном површином, користе се *отворени гранични услови* (енг. *opening boundary conditions*), који дозвољавају двосмерени проток (улаз и излаз кроз исту површину) у зависности од локалних услова. Ови услови су кључни за нумеричку стабилност и физичку коректност при моделовању интеракције флуида са околином.

У општем случају, за сва поља постоје два типа граничних услова: *Нојманови* и *Дирихлеови*. За дато поље f , Нојманов (природни) гранични услов задаје нормални извод тог поља, тј. извод у правцу нормале $\mathbf{n}$, односно вредност његовог флукса на граници, тако да важи:

$$\frac{\partial f}{\partial \mathbf{n}} = g(\mathbf{x}), \quad (2.20)$$

где је $\mathbf{n}$ јединични вектор нормале на површину границе, а $g(\mathbf{x})$ задата функција. У случају када је $g(\mathbf{x}) = 0$ (хомогени Нојманов услов), поље f не варира у правцу нормале на границу. Стога се *услов без шока* (енг. *no-flux condition*) може сматрати специјалним случајем хомогеног Нојмановог услова.

Са друге стране, Дирихлеов гранични услов задаје директно вредност самог поља f на граници области у којој се врши прорачун (у нашем случају на површини зида):

$$f = g(\mathbf{x}). \quad (2.21)$$

Уколико је $g(\mathbf{x}) = 0$ и овај услов важи за векторско поље брзине $\mathbf{u}$, тада на граници имамо $\mathbf{u} = 0$. Ово представља математичку формулацију услова прилепљивања у вискозним флуидима, који описује чињеницу да се флуид „зауставља” на чврстој површини.

Глава 3

Нумеричке методе за решавање једначина кретања флуида

3.1 Лагранжов и Ојлеров приступ решавању једначина кретања флуида

У анализи континуума, као што су флуиди (течности и гасови) или чврста деформабилна тела у кретању, постоје два основна приступа: Лагранжов и Ојлеров [2, 5]. Лагранжов приступ третира континуум као систем честица: свака честица има свој тренутни положај $\mathbf{x}$ и брзину $\mathbf{u}$, а праћење њиховог кретања омогућава опис еволуције целог система. Насупрот томе, Ојлеров приступ, који је уобичајенији у анализи флуида, фокусира се на фиксне тачке у простору и прати како се физичке величине (нпр. брзина, притисак, густина) мењају у тим тачкама током времена. Овај приступ пружа природан оквир за моделовање протока у флуидима, нарочито када су кинематика протока и локалне промене истовремено важне.

Нумерички, Лагранжов приступ одговара систему честица, са или без мреже која их повезује, док Ојлеров приступ одговара коришћењу фиксне решетке (мреже) која остаје непромењена у простору, без обзира на кретање флуида. Кључ за повезивање ова два приступа је материјални извод [2]:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} + v \frac{\partial q}{\partial y} + w \frac{\partial q}{\partial z}, \quad (3.1)$$

где је q нека од величина које анализирамо, док u , v и w представљају компоненте брзине $\mathbf{u}$. Први члан, $\partial q / \partial t$, представља брзину којом се величина

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

q мења у фиксној тачки простора, што одговара чисто Ојлеровом приступу. Други члан, $\nabla q \cdot \mathbf{u}$, коригује вредност q због кретања флуида кроз ту фиксну тачку, што је основа Лагранжовог приступа.

Пренос физичке величине у флуиду под дејством векторског поља брзине $\mathbf{u}$ назива се *адвекција* (енг. *advection*). Једначина адвекције се заснива на материјалном изводу, а у најједноставнијем случају тај извод је једнак нули:

$$\frac{Dq}{Dt} = 0, \quad (3.2)$$

односно

$$\frac{\partial q}{\partial t} + \mathbf{u} \cdot \nabla q = 0, \quad (3.3)$$

што значи да је, у Лагранжовом приступу, вредност величине за сваку честицу константна током времена.

Лагранжов приступ је физички интуитиван јер природно описује кретање и еволуцију појединачних честица, што омогућава прецизно праћење њихове историје и олакшава визуелизацију процеса. Адвекција је аутоматски обухваћена кретањем честица, што поједностављује једначине. Овај приступ је посебно ефикасан за симулације са дисконтинуитетима, слободним површинама, сударима и раздвајањем материјала.

Недостаци Лагранжовог приступа су високи рачунарски захтеви при симулацији система са великим бројем честица, што ограничава његову примену. Уколико се користи у оквиру методе коначних елемената, велике деформације доводе до дисторзије мреже, што захтева њено поновно генерисање, или поново покретање симулације. Такође, овај приступ није ефикасан за стационарне проблеме, а задавање граничних услова је отежано, посебно на улазима, излазима и чврстим зидовима, јер честице могу напустити посматрани домен.

Ојлеров приступ карактерише коришћење фиксне рачунске мреже, што омогућава стабилну и ефикасну нумеричку имплементацију. Примењује се у динамици флуида за анализу поља брзине, притиска и температуре, уз прецизно задавање граничних услова на зидовима, улазима и излазима. Овај приступ, такође, представља основу за моделовање турбулентних струјања.

Међутим, једна од главних мана Ојлеровог приступа је што не прати појединачне честице, већ поља у фиксираним тачкама. Ово отежава одређивање трајекторије или историје кретања одређених честица у случајевима када је то потребно. Нелинеарни адвективни члан који се појављује у динамичким једначинама кретања флуида често узрокује нумеричке проблеме, као што су

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

нестабилност и осцилације, што захтева коришћење сложенијих дискретизационих шема. Праћење слободних површина и међуповршина флуида захтева примену додатних нумеричких метода. Тачност решења зависи од квалитета, густине и структуре рачунске мреже, јер неадекватна мрежа узрокује нетачне или нестабилне резултате.

У овом раду углавном примењујемо Ојлеров приступ из следећих разлога:

- *Аналитичка једноставност*: природније је и лакше радити са просторним изводима, као што су градијент притиска или вискозни члан;
- *Једноставност нумеричке апроксимације*: значајно је лакше апроксимирати те изводе на фиксној мрежи него на „облаку“ произвољно покретних честица.

3.2 Дискретизација флуидних поља

Класификација метода

Дискретизација представља кључан корак у нумеричком решавању парцијалних диференцијалних једначина, као што су Навије-Стоксове једначине у динамици флуида. Циљ дискретизације је трансформација континуалног математичког модела у систем алгебарских једначина који се може ефикасно решити помоћу рачунара. Овај процес подразумева поделу домена симулације на коначан број подобласти, у којима се непознате величине (као што су брзина, притисак и температура) апроксимирају коришћењем одговарајућих нумеричких шема. Најчешће коришћене методе дискретизације су: *метода коначних разлика* (енг. *finite difference method* - FDM), *метода коначних запремина* (енг. *finite volume method* - FVM) и *метода коначних елемената* (енг. *finite element method* - FEM) [5]. Избор одговарајуће методе зависи од природе физичког проблема, геометрије домена, захтева за тачношћу и доступних рачунских ресурса.

FDM је једна од најстаријих и најједноставнијих техника за нумеричко решавање диференцијалних једначина, заснована на дискретизацији домена помоћу структуриране прорачунске мреже. Лака имплементација и ефикасност за базичне проблеме чине је погодном за образовне сврхе. Недостаци укључују ограничења при решавању проблема са сложенom геометријом (закривљене површине), као и сложеност у нумеричкој имплементацији нелине-

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

Табела 3.1: Решавачи за симулацију флуида у рачунарској графици.

Назив решавача	Тип лиценце	Примарна нумеричка метода	Примарна намена у рачунарској графици
Mantaflow [18]	Open Source	Ојлеров метод, метод ниво скупа, FLIP	Стандард за академска истраживања и нискобуџетну 3D продукцију.
Fluid Engine Dev [19]	Open Source	Лагранжов метод, метод ниво скупа, FLIP, SPH	C++ оквир (framework); директна имплементација алгоритама рачунарске графике из литературе.
SPlisHSPlasH [20]	Open Source	Лагранжов метод, SPH	Симулација вискозне динамике, површинског напона и интеракције честица у домену рачунарске графике.
Taichi [21]	Open Source	Лагранжов метод, MPM, FLIP, SPH (GPU убрзање)	Развој високоперформансних решавача за савремена истраживања.
SideFX Houdini [22]	Комерцијални	Ојлеров метод, FLIP	Индустријски стандард за визуелне ефекте (VFX) и анимацију.
Next Limit RealFlow [23]	Комерцијални	Лагранжов метод, FLIP	Висококвалитетна симулација флуида и интеракције са чврстим телима за филм.
Chaos Phoenix [24]	Комерцијални	Ојлеров метод	Брза и стабилна симулација дима, ватре и течности.
Autodesk Bifrost [25]	Комерцијални	Лагранжов метод, FLIP	Симулација флуида и аеродинамике директно у Autodesk Maya окружењу.
JangaFX EmberGen [26]	Комерцијални	Ојлеров метод (GPU убрзање)	Симулација дима и ватре за игре и виртуелну продукцију.
Obi Fluid [27]	Комерцијални	Лагранжов метод, SPH	Симулација флуида у реалном времену.

арних граничних услова. Такође, у тродимензионалним симулацијама, јавља се „проклетство димензионалности” (кубни раст броја чворова мреже дуж једне димензије), што драматично повећава рачунске и меморијске захтеве, док нумеричка дифузија и осцилације ограничавају тачност. Упркос томе, FDM остаје популарна техника у рачунарској графици за симулацију флуида и визуелних ефеката попут дима, пламена, и таласања воде.

FVM је кључна нумеричка метода у рачунарској динамици флуида, заснована на законима очувања масе, количине кретања и енергије. Обезбеђује

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

строгу конзервативност, гарантујући важење наведених закона на нивоу сваке контролне запремине, независно од густине мреже. FVM је посебно погодна за неструктуриране мреже и сложене геометрије, што је чини робусном и флексибилном у индустријским применама. Имплементирана је у најпознатијим софтверским пакетима за област динамике флуида, као што су ANSYS Fluent и OpenFOAM. Иако је имплементација методе FVM сложенија у поређењу са FDM, а основне шеме могу имати нижи ред тачности, могуће је побољшати тачност коришћењем апроксимација вишег реда.

FEM представља напредну технику која се ослања на партиционисање домена на мање елементе, унутар којих се решење апроксимира коришћењем локализованих базних функција. Омогућава ефикасну анализу сложених геометрија и неструктурираних мрежа, а посебно се истиче својим високим редом тачности, што се може додатно побољшати применом изопараметарских елемената. Широко се примењује у инжењерству, укључујући симулације стишљивих и нестишљивих струјања. Међутим, FEM се одликује сложенијом математичком и програмском имплементацијом, а очување физичких конзервативних својстава, као што су маса и импулс, захтева додатне модификације модела, што може закомпликовати процес анализе.

У Табели 3.1 дат је преглед најчешће коришћених решавача за симулацију флуида у рачунарској графици. Међу софтверима отвореног кода, Mantaflow [18] се истиче као референтни стандард за Ојлеров приступ и метод ниво скупа при моделовању слободних површина. С друге стране, Fluid Engine Dev [19] нуди C++ оквир за имплементацију FLIP (енг. *Fluid Implicit Particle*), као и Ојлеровог приступа и метода ниво скупа. SPliHSPlasH [20] је водећи алат за SPH (енг. *Smoothed Particle Hydrodynamics*) методу, док Taichi [21] омогућава развој напредних MPM (енг. *Material Point Method*) и FLIP решавача уз GPU (енг. *Graphics Processing Unit*) убрзање. Комерцијални алати попут SideFX Houdini [22] и RealFlow [23] и даље доминирају у високобуџетној VFX продукцији.

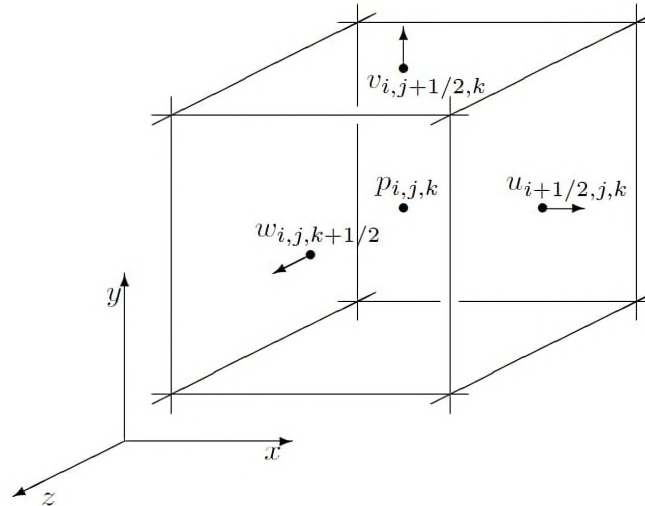
Врсте прорачунских мрежа за методу коначних разлика

Структуриране мреже у динамици флуида представљају регуларне дискретне структуре где сваки чвор мреже има фиксиран број суседа и познате координате. Најчешће се користе правоугаоне (Декартове) мреже код којих су контролне запремине организоване у регуларну ћелијску решетку са кон-

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

стантним корацима дискретизације Δx , Δy и Δz . Ово омогућава директну примену класичних шема коначних разлика као што су централне разлике за диференцијалне операторе и *шема узводних разлика* (енг. *upwind*) за конвективне чланове. Неравномерне структуриране мреже, са згуснутим ћелијама према зиду, омогућавају прецизно разрешавање граничних слојева уз минимално повећање укупног броја ћелија. Блок-структуриране мреже комбинују више структурираних блокова за апроксимацију сложенијих геометрија. Ове мреже омогућавају ефикасну имплементацију алгебарских операција и оптимизоване нумеричке шеме због регуларне структуре која омогућава коришћење оптимизованих алгоритама и векторских операција.

Мреже на којима се реализују тродимензионални прорачуни методом коначних разлика могу се класификовати на различите начине [28]. Оне намењене за дискретизацију скаларних величина деле се према положају података у ћелији на: *запремински центриране* (енг. *cell centered*), *чворно центриране* (енг. *vertex centered*) и *слински центриране* (енг. *face centered*). Мреже намењене прорачунима векторских величина могу бити *колоциране* (енг. *collocated*) и ова група се дели на запремински центриране и чворно центриране. За нестишљиви проток флуида користе се специјализоване мреже као што је *померена мрежа* (енг. *staggered*), приказана на Слици 3.1.



Слика 3.1: Ћелија померене мреже [2].

Померена мрежа подразумева специфичан распоред променљивих у оквиру рачунских ћелија са циљем побољшања нумеричке стабилности и прецизно-

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

сти приликом дискретизације једначина струјања. Притисак је дефинисан у центру сваке ћелије, док се компоненте брзине смештају у центре одговарајућих страна: x -компонента (u) на стране нормалне на x -осу, y -компонента (v) на стране нормалне на y -осу, а z -компонента (w) на стране нормалне на z -осу. Овај распоред назива се померена мрежа јер су компоненте брзине и притисак смештени у различитим тачкама у оквиру исте рачунске ћелије. Такав приступ омогућава бољу нумеричку стабилност и ефикасније задовољење услова нестишљивости ($\nabla \cdot \mathbf{u} = 0$), што га чини посебно погодним за симулацију тока флуида. Померена мрежа је први пут примењена у оквиру *MAC* (енг. *Marker-and-Cell*) методе, која се сматра почетком савремене нумеричке динамике флуида (енг. *Computational Fluid Dynamics* - CFD) [29].

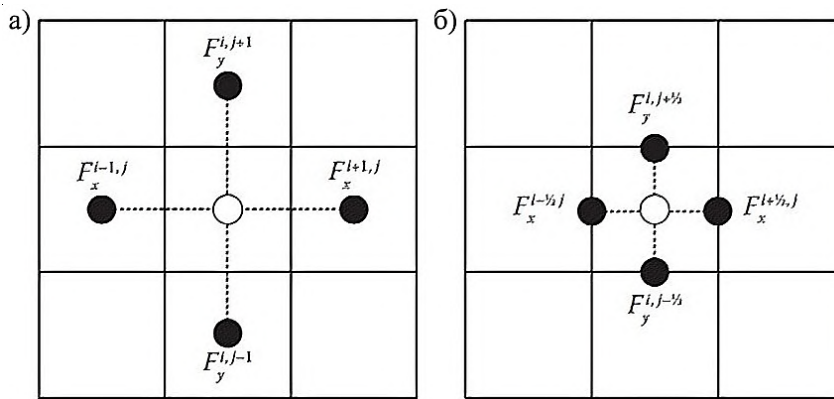
Применом централних разлика на запремински центрираним колоцираним мрежама добија се:

$$\nabla \cdot \mathbf{F}(\mathbf{x}) \approx \frac{F_x^{i+1,j,k} - F_x^{i-1,j,k}}{2\Delta x} + \frac{F_y^{i,j+1,k} - F_y^{i,j-1,k}}{2\Delta y} + \frac{F_z^{i,j,k+1} - F_z^{i,j,k-1}}{2\Delta z}. \quad (3.4)$$

На основу чињенице да централна разлика захтева само податке суседних тачака, а не саму централну тачку, претходну једначину можемо модификовати тако да буде погодна за коришћење на помереним мрежама:

$$\nabla \cdot \mathbf{F}(\mathbf{x}) \approx \frac{F_x^{i+\frac{1}{2},j,k} - F_x^{i-\frac{1}{2},j,k}}{\Delta x} + \frac{F_y^{i,j+\frac{1}{2},k} - F_y^{i,j-\frac{1}{2},k}}{\Delta y} + \frac{F_z^{i,j,k+\frac{1}{2}} - F_z^{i,j,k-\frac{1}{2}}}{\Delta z}, \quad (3.5)$$

где се као корак користи половина размака између тачака мреже.



Слика 3.2: Израчунавање дивергенције на а) колоцираним и б) помереним мрежама [28].

На Слици 3.2 илустровано је израчунавање дивергенције на а) колоцираним и б) помереним мрежама. Ради једноставности приказан је дводимензи-

онални случај. Код померених мрежа, сви подаци потребни за израчунавање припадају једној ћелији и нема потребе за посебном обрадом граничних случајева. Ово резултира повећањем ефективне резолуције, односно смањењем ефективне величине мреже, што може побољшати тачност прорачуна. Због тога су померене мреже оптималан избор за дискретизацију векторских поља у прорачунима са учесталом применом оператора дивергенције. На помереним мрежама, Лапласов оператор се дискретизује применом централних разлика:

$$\Delta f(\mathbf{x}) \approx \frac{f^{i+1,j,k} - 2f^{i,j,k} + f^{i-1,j,k}}{\Delta x^2} + \frac{f^{i,j+1,k} - 2f^{i,j,k} + f^{i,j-1,k}}{\Delta y^2} + \frac{f^{i,j,k+1} - 2f^{i,j,k} + f^{i,j,k-1}}{\Delta z^2}. \quad (3.6)$$

Овај облик ће бити од користи при израчунавању притиска.

3.3 Метода раздвајања оператора

Метода раздвајања оператора (енг. *operator splitting*) представља основни алат у нумеричком моделовању којим се сложени проблеми разлажу на низ једноставнијих потпроблема [2, 5]. Многи физички процеси, као што су адвекција и дифузија, обухватају више механизма описаних различитим математичким операторима. Пошто су једначине са појединачним операторима знатно лакше за решавање, за њих често постоје високо оптимизовани, па и хардверски убрзани алгоритми. Овај приступ омогућава примену специјализоване нумеричке методе за сваки оператор посебно, чиме се унапређују прецизност и ефикасност прорачуна сложених система.

Размотримо Кошијев проблем облика:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{L}(\mathbf{u}) = (\mathbf{A} + \mathbf{B})\mathbf{u} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{u}, \quad (3.7)$$

где је $\mathbf{u}$ непозната функција, $\mathbf{L}$ је линеарни оператор, док су $\mathbf{A}$ и $\mathbf{B}$ његове компоненте. Најједноставнији облик раздвајања је *Лије-Тројерово секвенцијално раздвајање*, које се заснива на секвенцијалном решавању потпроблема [30]. У првом кораку решава се проблем:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{A}\mathbf{u}, \quad (3.8)$$

на временском интервалу $[t, t + \Delta t]$, са почетним условом $\mathbf{u}(t) = \mathbf{u}^n$, чиме се добија међурешење $\mathbf{u}^*$. Затим се решава једначина:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{B}\mathbf{u}, \quad (3.9)$$

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

на истом интервалу, али са почетним условом $\mathbf{u}(t) = \mathbf{u}^*$ и добија се коначно решење $\mathbf{u}^{n+1}$. Алгоритам, којим се из старог стања $\mathbf{u}^n$ добија ново стање $\mathbf{u}^{n+1}$ преко низа једноставнијих корака, симболички је дат формулом:

$$\mathbf{u}^{n+1} = \exp(\Delta t \mathbf{B}) \circ \exp(\Delta t \mathbf{A})(\mathbf{u}^n). \quad (3.10)$$

Уколико оператори $\mathbf{A}$ и $\mathbf{B}$ нису комутативни, наведена релација важи као апроксимација за довољно мале вредности временског корака Δt . Овај приступ је једноставан за имплементацију и добро се уклапа у постојеће нумеричке шеме, али има грешку првог реда по времену $O(\Delta t)$ (грешка је сразмерна Δt). Због тога се често користи у ситуацијама где је брзина важнија од високе тачности.

За бољу тачност користи се *Сиренцово раздвајање* које спада у групу симетризованих шема [31]. У овом приступу, прво се решава једначина (3.8) на интервалу $[t, t + \Delta t/2]$, са почетним условом $\mathbf{u}(t) = \mathbf{u}^n$ и добија се решење $\mathbf{u}^*$. Затим се решава једначина (3.9) на комплетном интервалу $[t, t + \Delta t]$ и добија решење $\mathbf{u}^{**}$. Као последњи корак решава се поново једначина (3.8) на интервалу $[t + \Delta t/2, t + \Delta t]$, са почетним условом $\mathbf{u}(t + \Delta t/2) = \mathbf{u}^{**}$ што даје коначан резултат $\mathbf{u}^{n+1}$.

Метода раздвајања оператора доноси бројне предности у нумеричком моделовању. Прво, ова метода поједностављује имплементацију јер омогућава коришћење постојећих, добро тестираних кодова за решавање проблема описаних појединачним операторима. Друго, повећава ефикасност, будући да су алгоритми за појединачне операторе, као што су метода Рунге-Кута за чисту адвекцију и имплицитне шеме за чисту дифузију, често бржи и стабилнији у поређењу с комбинованим операторима. Треће, ова метода омогућава да се за сваки оператор може изабрати оптимална нумеричка шема - експлицитна за брзе процесе и имплицитна за споре или стабилне процесе. Коначно, модуларност методе чини додавање новог оператора релативно једноставним.

Приликом нумеричког решавања Навије-Стоксових или Ојлерових једначина за нестишљиве флуиде, један од најзначајнијих изазова је раздвајање спрегнутости између поља брзине и притиска, која произилази из услова нестишљивости $\nabla \cdot \mathbf{u} = 0$. Овај услов, који захтева да векторско поље брзине буде недивергентно (дивергенција једнака нули), уводи нелокалну зависност притиска од брзине и чини директну имплементацију Навије-Стоксових једначина нумерички захтевном. Метода раздвајања оператора овде пружа елегантно и ефикасно решење. Уместо да се сви физички процеси решавају

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

истовремено у спрегнутом систему, они се раздвајају на низ једноставнијих, физички интуитивних корака који се решавају секвенцијално. Овај приступ не само што поједностављује алгоритамску имплементацију, већ омогућава и коришћење специјализованих нумеричких шема за сваки појединачни процес. У овом раду нећемо узимати у обзир утицај вискозних сила, па стога нећемо ни наводити релације које то омогућавају. Дакле, у случају Ојлерових једначина, раздвајање ћемо извршити на три компоненте: адвекцију, деловање спољашњих сила и корекцију притиска [2].

- Адвекцијски корак:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \mathbf{u} \nabla q = 0. \quad (3.11)$$

У овом кораку се израчунава интермедијарно поље брзине при чему се занемарују члан притиска и услов нестишљивости. Овде је употребљена ознака q у једначини адвекције јер нас може интересовати адвекција и других поља, а не само брзине $\mathbf{u}$.

- Деловање спољашњих сила:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{g}. \quad (3.12)$$

Брзине добијене у адвекцијском кораку се коригују уз претпоставку да је гравитација једина спољашња сила у систему. Као резултат прва два корака добијамо интермедијарно поље брзина $\mathbf{u}^*$.

- Корекција притиска: Интермедијарно поље брзина се пројектује на потпростор недивергентних поља увођењем корекције притиска, чиме се задовољава услов нестишљивости. Коначно поље $\mathbf{u}$ се добија из дискретизоване релације:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla p, \quad (3.13)$$

где је p кориговани притисак чиме је обезбеђено да важи $\nabla \cdot \mathbf{u} = 0$.

Обједињавањем ових корака, добија се основни алгоритам за симулацију струјања флуида:

- Полази се од почетног, недивергентног поља брзине $\mathbf{u}^0$;
- За сваки прелазак $t_n \rightarrow t_{n+1}$ бира се временски корак Δt који задовољава услове нумеричке стабилности на целом интервалу (t_0, t_N) .

За сваки временски корак n извршава се:

- адвекција: $\mathbf{u}^A = \text{advect}(\mathbf{u}^n, \Delta t, \mathbf{u}^n)$,
- урачунавање деловања запреминских сила: $\mathbf{u}^B = \mathbf{u}^A + \Delta t \mathbf{g}$,

- пројекција притиска: $\mathbf{u}^{n+1} = \text{project}(\Delta t, \mathbf{u}^B)$.

У наредним пасусима ћемо детаљније описати кораке адвекције и пројекције притиска.

3.4 Семи-Лагранжова адвекција

У рачунаској графици се готово искључиво користи семи-Лагранжова метода у адвекцијском кораку прорачуна, па ће стога она овде бити детаљно описана [2]. Напоменимо да се у ојлеровским флуидним прорачунима користе и друге методе (*upwind*, WENO, QUICK, итд.). Детаљни опис ових метода може се наћи у бројним уџбеницима и монографијама посвећеним нумеричким прорачунима у области динамике флуида [5]. Разлози за примену семи-Лагранжовог поступка за решавање адвекционе једначине најједноставније се могу илустровати на примеру у једној димензији:

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} = 0. \quad (3.14)$$

Најпре ћемо изводе заменити коначним разликама. Ако се за временски извод користи експлицитна Ојлерова шема, а за просторни извод централна разлика другог реда тачности, из горње једначине добијамо:

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + u_i^n \frac{q_{i+1}^n - q_{i-1}^n}{2\Delta x} = 0. \quad (3.15)$$

Једнакост (3.15) можемо преуредити у експлицитну формулу за израчунавање нових вредности величине q , на основу претходних:

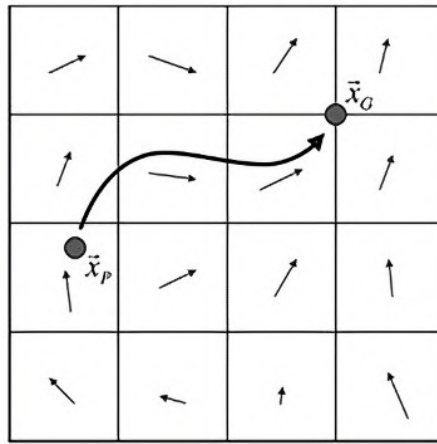
$$q_i^{n+1} = q_i^n - \Delta t u_i^n \frac{q_{i+1}^n - q_{i-1}^n}{2\Delta x}. \quad (3.16)$$

Ова шема је једноставна за имплементацију, али има значајне проблеме са нумеричком стабилношћу. Наиме, Ојлерова метода је нестабилна када се комбинује са централном разликом за дискретизацију просторног извода, без обзира на то колико мали временски корак Δt се користи и нумеричко решење неминовно дивергира. Може се показати да се у решењу појављују неприродне високофреквентне осцилације које се током времена акумулирају, чинећи поступак решавања дивергентним.

Семи-Лагранжов метод, којег је у рачунарску графику увео Стем [32], је једноставан и физички заснован приступ адвекцији, утемељен на релацији

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

$Dq/Dt = 0$, према којој се особина честице q не мења дуж њене трајекторије. Да бисмо одредили вредност q у некој тачки $\mathbf{x}$ у тренутку $t + \Delta t$, довољно је да пронађемо честицу која у том тренутку стиже у $\mathbf{x}$. Уместо да пратимо честице унапред у времену (тј. у смеру протицања времена), пратимо положај фиктивне честице која се у тренутку $t_{n+1} = t_n + \Delta t$ налази у чвору мреже (означеном са $\mathbf{x}_G$) уназад кроз време - односно, трагамо за њеним положајем у претходном тренутку t_n . С обзиром на то да њен стари положај не мора одговарати неком од мрежних чворова, вредност величине q у том положају означимо са q_P^n , и она се одређује интерполацијом на основу њених вредности у суседним чворовима. Нова вредност q у $\mathbf{x}_G$ је онда $q_G^{n+1} = q_P^n$. Овај приступ је стабилан, има физички смисао и често се користи у симулацијама флуида и атмосферских модела.



Слика 3.3: Кретање фиктивне честице [28].

Најпре је потребно одредити почетну позицију фиктивне честице, означену са $\mathbf{x}_P$. Кретање честице описује се диференцијалном једначином $d\mathbf{x}/dt = \mathbf{u}(\mathbf{x})$, тако да она након временског интервала Δt стиже у тачку $\mathbf{x}_G$, као што је приказано на Слици 3.3. Ако променимо смер протока времена, можемо ићи у супротном смеру од тачке $\mathbf{x}_G$ до почетне позиције честице тј. можемо пронаћи где би честица завршила у инверзном пољу брзине $-\mathbf{u}$. Најједноставнији начин да се процени $\mathbf{x}_P$ јесте примена једног корака експлицитног Ојлеровог метода, али уназад у времену:

$$\mathbf{x}_P = \mathbf{x}_G - \Delta t \cdot \mathbf{u}(\mathbf{x}_G), \quad (3.17)$$

где користимо вредност брзине $\mathbf{u}$ у чвору мреже да бисмо направили корак дужине Δt уназад кроз поље струјања. Експлицитна Ојлерова метода

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

понекад може бити довољна, али се знатно бољи резултати могу постићи коришћењем методе Рунге–Кута (РК) вишег реда. Метода Рунге–Кута другог реда (РК2) дефинисана је са:

$$\mathbf{x}_{\text{mid}} = \mathbf{x}_G - \frac{1}{2}\Delta t \cdot \mathbf{u}(\mathbf{x}_G), \quad (3.18)$$

$$\mathbf{x}_P = \mathbf{x}_G - \Delta t \cdot \mathbf{u}(\mathbf{x}_{\text{mid}}). \quad (3.19)$$

Након полукоракa (3.18) добија се међуконфигурација $\mathbf{x}_{\text{mid}}$, која апроксимира положај честице на средини трајекторије. Поље брзина се затим интерполира са мреже у овој међупозицији, а добијена вредност брзине користи се за коначно одређивање положаја $\mathbf{x}_P$. Још бољи резултати, посебно за вртлоге, постижу се коришћењем методе Рунге–Кута четвртог реда (РК4).

Како се $\mathbf{x}_P$ ретко поклапа са чвором мреже, вредност q се мора одредити интерполацијом на основу вредности q^n на суседним чворовима мреже (билинеарно у 2D, односно трилинеарно у 3D). Иако је семи-Лагранжова метода, захваљујући праћењу уназад, робустна за велике временске кораке, она је прецизна само при високој резолуцији. У супротном настаје грешка апроксимације, слична оној код Ојлерове временске интеграције. Уколико грешка временске интеграције доведе до одступања нумеричког решења од тачне трајекторије струјања, грешка просторне интерполације уноси додатну нумеричку дисипацију, познату као *нумеричка дифузија*. При адвекцији густине, овај феномен замагљује локалне детаље и доводи до губитка масе, док код поља брзине уноси вештачку вискозност која пригушује вртлоге и умањује визуелну реалистичност симулације.

Постоје технике које замењују линеарну интерполацију методама вишег реда, као што су интерполација кубним полиномима (сплајнови), нпр. Катмул–Ромова метода [28]. Међутим, интерполације вишег реда нарушавају једну од кључних карактеристика оригиналне семи-Лагранжове методе. Наиме, линеарна интерполација користи линеарну функцију за апроксимацију, која је монотона. Због тога интерполација гарантује да ће резултат остати у оквиру минималне и максималне вредности оригиналног поља. Кубна интерполација не гарантује монотоност, што значи да интерполирана вредност може изаћи ван овог опсега и може доћи до *прекорачења* (енг. *overshoots*).

3.5 Пројекција притиска

Интермедијарно поље брзина $\mathbf{u}^*$, које се добија после адвекционог корака и урачунавања ефеката спољашњих сила, не задовољава услов нестишљивости $\nabla \cdot \mathbf{u}^* = 0$. Овај проблем се може решити израчунавањем поља притиска које ће деловати на поље брзина тако да га учини недивергентним. У стварном флуиду, притисак се тренутно мења како би спречио било какву локалну промену дивергенције. Један од начина да се поље брзина учини недивергентним, истовремено са одређивањем одговарајућег поља притиска, јесте коришћење *метода релаксације*. У овом приступу, итеративно се коригују поља притиска и брзине све док се не постигне поље брзина са занемаривом дивергенцијом. На крају се поље брзина $\mathbf{u}^*$ ажурира у складу с променама притиска.

Из релације (3.13) за брзину $\mathbf{u}^*$ добија се једначина:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \frac{\nabla p}{\rho}, \quad (3.20)$$

у којој претпостављамо да је густина ρ константна, а притисак p непознат. Временски корак Δt има такође константну вредност у свим ћелијама мреже. Услов нестишљивости се може написати у облику:

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \quad (3.21)$$

Како смо већ рекли, $\nabla \cdot \mathbf{u}^*$ не мора бити једнако нули. Међутим, $-\nabla p$ је сила која ће елиминисати онај део брзине који ствара дивергенцију, тако да важи (3.21), односно да коначно поље брзина буде недивергентно. На основу ове претпоставке, применимо оператор дивергенције на једначину (3.20). Тада се једначина трансформише у:

$$\nabla \cdot \mathbf{u}^{n+1} = \nabla \cdot \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla \cdot \nabla p. \quad (3.22)$$

На основу (3.21), лева страна постаје једнака нули. Користећи дефиницију Лапласовог оператора, $\nabla \cdot \nabla = \Delta$, једначина (3.22) се може написати у облику:

$$\frac{\Delta t}{\rho} \Delta p = \nabla \cdot \mathbf{u}^*. \quad (3.23)$$

На овај начин добија се Пуасонова једначина чијим се решавањем добија непозната вредност притиска p . Она се може дискретизовати методом коначних разлика на помереној прорачунској мрежи описаној у одељку 3.2. Диференцијску шему која одговара једначини (3.23) најједноставније је добити

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

полазећи од шеме којом се дискретизује једначина (3.20):

$$u_{i+1/2,j,k}^{n+1} = u_{i+1/2,j,k}^* - \Delta t \frac{1}{\rho} \frac{p_{i+1,j,k} - p_{i,j,k}}{\Delta x}, \quad (3.24)$$

$$v_{i,j+1/2,k}^{n+1} = v_{i,j+1/2,k}^* - \Delta t \frac{1}{\rho} \frac{p_{i,j+1,k} - p_{i,j,k}}{\Delta y}, \quad (3.25)$$

$$w_{i,j,k+1/2}^{n+1} = w_{i,j,k+1/2}^* - \Delta t \frac{1}{\rho} \frac{p_{i,j,k+1} - p_{i,j,k}}{\Delta z}, \quad (3.26)$$

а затим ове изразе уврстити у дискретизовану форму услова нестишљивости (3.23), коришћењем (3.5):

$$\frac{u_{i+1/2,j,k}^{n+1} - u_{i-1/2,j,k}^{n+1}}{\Delta x} + \frac{v_{i,j+1/2,k}^{n+1} - v_{i,j-1/2,k}^{n+1}}{\Delta y} + \frac{w_{i,j,k+1/2}^{n+1} - w_{i,j,k-1/2}^{n+1}}{\Delta z} = 0. \quad (3.27)$$

На тај начин се долази до финалног облика диференцијалне једначине којом се дискретизује једначина (3.23):

$$\begin{aligned} & \frac{(p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k})}{\Delta x^2} + \frac{(p_{i,j+1,k} - 2p_{i,j,k} + p_{i,j-1,k})}{\Delta y^2} + \frac{(p_{i,j,k+1} - 2p_{i,j,k} + p_{i,j,k-1})}{\Delta z^2} = \\ & = \frac{\rho}{\Delta t} \left[\frac{(u_{i+1/2,j,k}^* - u_{i-1/2,j,k}^*)}{\Delta x} + \frac{(v_{i,j+1/2,k}^* - v_{i,j-1/2,k}^*)}{\Delta y} + \frac{(w_{i,j,k+1/2}^* - w_{i,j,k-1/2}^*)}{\Delta z} \right]. \end{aligned} \quad (3.28)$$

Ако се ћелија прорачунске мреже налази на граници домена, нове брзине на граничним површинама зависе од притиска изван домена, који се задаје граничним условима. Једначином (3.28) дефинисан је систем линеарних једначина чијим се решавањем добијају вредности притиска $p_{i,j,k}$ у свим ћелијама мреже. Овај систем можемо представити у облику матричне једначине: на левој страни једначине налази се матрица коефицијената $\mathbf{A}$ помножена вектором који садржи све непознате вредности притиска $\mathbf{p}$, док је на десној страни вектор дивергенција интермедијарних брзина $\mathbf{b}$ (са одговарајућим изменама на границама чврстих зидова):

$$\mathbf{A}\mathbf{p} = \mathbf{b}. \quad (3.29)$$

Свака врста матрице $\mathbf{A}$ одговара једној једначини, тј. једној ћелији рачунске мреже. На пример, ако је ћелија мреже (i, j, k) испуњена флуидом, постојаће врста матрице коју можемо означити индексима (i, j, k) . Елементи те врсте су коефицијенти свих непознатих притисака у тој једначини: готово сви они су нула, осим можда седам елемената који одговарају притисцима

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

$p_{i,j,k}$ и његових шест суседа, $p_{i\pm 1,j,k}$, $p_{i,j\pm 1,k}$ и $p_{i,j,k\pm 1}$. Ненулни елементи постоје само за (i, j, k) и његове суседе који су такође ћелије флуида. Дакле, у питању је ретка матрица која је поред тога и симетрична, што олакшава решавање једначине (3.29).

Методe за решавање система линеарних једначина, посебно оних који настају приликом дискретизације парцијалних диференцијалних једначина, могу се поделити на итеративне и директне методе. Избор одговарајуће методе зависи од карактеристика проблема, као што су величина система, структура матрице, симетричност, условљеност и доступни рачунски ресурси. Метода конјугованог градијента (енг. *Conjugate Gradient* – CG) је једна од најпопуларнијих итеративних метода, поготово за решавање система са симетричном позитивно дефинитном матрицом [33]. Ова метода се истиче малим меморијским захтевима и ефикасношћу, што је чини погодном за велике проблеме који настају у научним прорачунима. Како би се побољшала брзина конвергенције, метода се често комбинује са прекондиционерима, као што су Јакобијев прекондиционер или непотпуна Чолескијева декомпозиција. Мултигрид методе представљају још једну високо ефикасну класу алгоритама, посебно за решавање Поасонове једначине и других елиптичких проблема [33]. Мултигрид приступ се може користити као самостални решавач или као веома ефикасан прекондиционер у оквиру других итеративних метода, попут CG методе, што значајно повећава укупну брзину решавања.

Када матрица система није симетрична, што је чест случај у нестационарним или нелинеарним проблемима, метода конјугованог градијента није применљива. У таквим ситуацијама, често се користи *BiCGSTAB* (енг. *Bi-Conjugate Gradient Stabilized*) метода, који је стабилнија и поуздана варијанта BiCG методе [33]. Захваљујући побољшаној нумеричкој стабилности и бољој конвергенцији, BiCGSTAB је постао стандард у решавању несиметричних система, посебно у контексту динамике флуида и комплексних физичких модела.

За опште несиметричне системе, једна од најефикаснијих итеративних метода је *GMRES* (енг. *Generalized Minimal Residual*), позната по високој прецизности и сигурној конвергенцији [33]. Међутим, њен главни недостатак је повећана потрошња меморије и рачунских ресурса, јер се са сваком итерацијом морају чувати сви претходни вектори. Због тога се често користи у модификованом облику са рестартовањем (*GMRES(m)*), како би се ограни-

чило заузеће меморије.

Коначно, директне методе обезбеђују тачно решење система и користе се у ситуацијама када је кључна висока прецизност. Међутим, за велике тродимензионалне проблеме, ове методе постају рачунски превише захтевне због квадратне или чак кубне сложености по величини система. Стога се директне методе обично ограничавају на мање проблеме, специјализоване примене или ситуације у којима се матрица система не мења током времена (нпр. стационарни проблеми са фиксном мрежом). У пракси, за велике системе, итеративне методе са одговарајућим прекондиционерима се чешће користе од директних.

У рачунарској графици, где је брзина често важнија од апсолутне тачности, најчешће се користе итеративне методе, посебно метода конјугованог градијента у комбинацији са Јакобијевим или мултигрид прекондиционерима. Ове комбинације омогућавају одличан баланс између брзине извршавања, стабилности и једноставности имплементације, што је кључно за апликације у реалном времену као што су симулације течности и анимације.

3.6 Стабилност прорачуна

Један од највећих проблема сваке нумеричке методе је њена стабилност: да ли ће грешке накупљене током прорачуна експоненцијално расти, што би довело до нестабилног и неупотребљивог решења. *CFL услов*, назван по математичарима Р. Куранту (R. Courant), К. Фридрихсу (K. Friedrichs) и Х. Левију (H. Lewy), је једноставан, интуитиван и потребан услов за конвергенцију нумеричког поступка. Под конвергенцијом се подразумева да, ако понављамо симулацију користећи све мање и мање временске кораке Δt и просторне кораке Δx , који у граничном случају теже нули, тада се решења добијена нумеричком методом морају све више приближавати тачном решењу. CFL услов тврди да временски корак Δt мора бити довољно мали у односу на просторну резолуцију Δx , тако да промена било које физичке величине (нпр. брзине, притиска) не прелази више од једне ћелије по временском кораку. На тај начин се осигурава да међусобна повезаност суседних тачака у мрежи буде адекватно заступљена у дискретној апроксимацији, што је неопходно за стабилност и тачност методе.

Семи-Лагранжов приступ је безусловно стабилан, јер решење не дивер-

ГЛАВА 3. НУМЕРИЧКЕ МЕТОДЕ ЗА РЕШАВАЊЕ ЈЕДНАЧИНА КРЕТАЊА ФЛУИДА

гира без обзира на величину временског корака Δt , што га чини посебно погодним за рачунарску графику и интерактивне симулације. Нова вредност физичке величине q у свакој тачки добија се праћењем трајекторије уназад и интерполацијом из вредности у околним тачкама у претходном временском кораку. Због линеарне интерполације, резултат увек остаје у опсегу вредности из претходног корака, спречавајући настанак нових екстремума. Ово омогућава да се временски корак Δt бира искључиво на основу жељеног компромиса између тачности и брзине прорачуна. На пример, за симулације у реалном времену, Δt може бити једнако трајању једног фрејма.

У пракси, ова метода може довести до нежељених резултата ако се величина временског корака превише повећа. Искуство показује да вредност Δt треба ограничити тако да се трајекторија честице прати највише на растојању од неколико, на пример пет, мрежних ћелија [2]:

$$\Delta t \leq \frac{5\Delta x}{u_{\max}}, \quad (3.30)$$

где је $u_{\max}$ процена максималне брзине у флуиду. Ова вредност може, на пример, бити највећа брзина тренутно израчуната на мрежи. Робустнија процена узима у обзир и брзине које могу настати због убрзања $\mathbf{g}$ изазваног гравитацијом (или других запреминских сила) током временског корака.

Ипак, и при оваквом временском кораку могу настати нежељени артефакти. Да би се избегло рачунски захтевно поновно покретање симулације са мањим кораком, трајекторије у семи-Лагранжовој адвекцији се могу пратити кроз више мањих поткорака. Ограничавањем сваког поткорака условом $|\mathbf{u}(\mathbf{x})|\Delta t < \Delta x$ (померај до једне ћелије мреже), вероватноћа настанка грешака постаје занемарљива. Ово ограничење се примењује локално: бржи региони флуида захтевају више поткорака, чиме се одржава стабилност уз максималну рачунску ефикасност.

Неке методе, попут комбинације напредног Ојлеровог метода и централних разлика, су нестабилне без обзира на величину временског корака. С друге стране, постоје експлицитне методе стабилне за произвољно велике временске кораке, али оне не конвергирају ка тачном решењу без испуњеног CFL услова. Дакле, CFL услов је неопходан за конвергенцију, док стабилност само спречава неограничен раст грешака.

Глава 4

Двофазни ток флуида

Двофазни ток флуида је појава која настаје када се у истом систему истовремено налазе две фазе исте супстанце, најчешће течна и гасовита, или две различите течности које се не мешају. Такви токови се јављају у природи (таласи на површини воде, активности гејзира и ширење дима), али и у индустријским процесима, што захтева развој прецизних математичких модела за њихово описивање. За нас су од посебног интереса примене двофазних токова у рачунарској графици, које омогућавају реалистичне симулације флуидних појава у филмским анимацијама, видео-играма и виртуелним окружењима.

Двофазни ток флуида је знатно сложенији од једнофазног због динамичких и термичких интеракција између фаза. У зависности од физичких услова, могу се формирати различити облици тока, попут мехурастог, капљичастог, прстенастог и стратификованог. Граница између фаза, позната као *гранична површина* или *међуповршина*, еволуира под утицајем динамике флуида. Један од циљева овог рада је моделовање и анализа временске еволуције међуповршине током двофазног тока, коришћењем векторског поља брзина добијеног нумеричким решавањем Навије–Стоксових једначина, детаљно описаним у претходној глави.

4.1 Дефиниција и класификација граничних површина између фаза

Постоје различити приступи симулацији покретних граничних површина између фаза, класификовани према начину њиховог представљања и праћења

у времену [34]. Нека је Ω просторни домен, а Γ граница у њему, димензије за један мање од Ω (тј. ко-димензије један), која дели домен на одвојене области. Граница је оштра ако је зона прелаза између фаза бесконачно танка, односно дифузна ако има коначну дебљину и гладак прелаз. Методе са оштром границом ефикасно прате форму, али се тешко имплементирају при тополошким променама. Насупрот томе, методе са дифузном границом боље обрађују комплексне тополошке промене, али захтевају високу резолуцију у зони границе, што је рачунски захтевно у одсуству адаптивне мреже.

Представљање граничне површине је кључно у нумеричким симулацијама. У Лагранжовим методама честице на међуповршини се крећу заједно с њом, па је њен положај експлицитно познат. Стога се ови приступи називају *методима праћења међуповршине* (енг. *interface tracking*) [2, 28]. Класичан пример Лагранжовог приступа, *метода маркер-честица* (енг. *Particle Method*), због бројних недостатака данас је у великој мери напуштена [2, 28]. Са друге стране, Ојлеров приступ користи фиксну мрежу, а међуповршина се представља као ниво (контура) имплицитно дефинисане функције. Такви приступи називају се *методима хватања међуповршине* (енг. *interface capturing*) [2, 28]. Постоје и хибридне методе које комбинују добре стране оба наведена приступа.

4.2 Гранична површина у двофазним токовима

У двофазном току, граница између фаза се може описати на различите начине, у зависности од тога да ли се систем посматра на микроскопском или макроскопском нивоу. На микроскопском (молекулском) нивоу, граница између фаза није оштра, већ постоји прелазна зона коначне дебљине. У тој зони се физичке особине система мењају од вредности карактеристичних за једну фазу до оних карактеристичних за другу. Ово указује на гладак, а не оштар прелаз између фаза.

На макроскопском нивоу, граница између фаза се описује као гранична површина која раздваја фазе са различитим особинама. На њој се јављају специфичне силе и ефекти, попут површинског напона, као и пренос масе и енергије између фаза. У овом раду следићемо искључиво макроскопски приступ у коме се граница третира као гранична површина, а њена промена

током времена се моделује помоћу одговарајућих нумеричких метода заснованих на механици континуума.

Као што је већ напоменуто, двофазни ток подразумева истовремено присуство две различите течности које се не мешају, или се врло слабо мешају. Типични примери таквих комбинација су уље и вода, односно вода и нека вискознија течност као што је глицерин. Посебно нас интересује двофазни ток у коме су фазе ваздух и течност (најчешће вода), који карактерише изузетно комплексна динамика границе између фаза, укључујући њено формирање, спајање и распад. Ове тополошке промене захтевају прецизно моделовање еволуције граничне површине, што представља један од главних изазова у симулацијама двофазних токова.

Површински напон се може занемарити када су инерцијалне силе знатно јаче од површинских, што је типично за системе већих димензија и високих брзина струјања. Међутим, код капиларних ефеката и процеса формирања капи, утицај површинског напона се не сме занемарити. У анализи двофазних токова користе се Навије-Стоксове једначине у којима параметри, као што су густина и вискозност, имају различите вредности у свакој од фаза. За праћење промена граничне површине између фаза током времена, развијен је низ нумеричких метода, међу којима су најзначајније:

- *Метода запреминских фракција* (енг. *Volume of Fluid Method - VOF*) користи дискретну функцију запреминске фракције α , која указује на удео одређене фазе у мрежној ћелији: $\alpha = 0$ означава прву фазу, $\alpha = 1$ другу, док вредности између 0 и 1 описују ћелије које садрже граничну површину. Ова метода обезбеђује добру очуваност масе и поуздано праћење границе, али са ограниченом тачношћу при израчунавању геометријских карактеристика као што су кривина и нормале [35].
- *Метода фазних поља* (енг. *Phase Field Method - PFM*) користи глатку функцију за опис танке прелазне зоне између фаза, чија се еволуција одређује минимизацијом одговарајућег Гинцбург-Ландауовог функционала. Овај приступ омогућава ефикасно моделовање сложених појава без експлицитног праћења граничне површине, али захтева строге услове равнотеже и високу резолуцију мреже у прелазној зони, што га чини рачунски захтевним [35].
- *Метода ниво скупа* (енг. *Level Set Method - LSM*) је једна од метода које користе репрезентацију међуповршине имплицитно дефинисаном

функцијом $\phi(\mathbf{x}, t)$ [13, 36]. Ова функција најчешће представља *означено растојање* (енг. *signed distance*) од међуповршине, док она сама одговара нултој изоповршини: $\phi(\mathbf{x}, t) = 0$. Имплицитни опис омогућава природно моделовање тополошких промена, као што су спајање и раздвајање фаза. Међутим, метода је склона нумеричкој дифузији, што може изазвати губитак очувања масе уколико се не користе додатне технике корекције [37].

Свака од наведених метода има своје предности и ограничења, а избор зависи од природе проблема и захтева примене. Метода VOF погодна је за проблеме са строгим захтевима за очување масе и јасно раздвојеним фазама. LSM се користи за проблеме са комплексним тополошким променама и кад су потребне прецизне геометријске информације о граничној површини. PFM се најчешће примењује у фундаменталним истраживањима са доминантним термодинамичким и мезоскопским ефектима. Због тога се у напредним симулацијама често користе хибридни приступи, који комбинују предности појединих метода ради постизања веће тачности, стабилности и рачунске ефикасности.

У рачунарској графици, VOF и LSM играју кључну улогу у креирању реалистичних симулација флуида. VOF се широко користи у филмској индустрији и видео-играма због тачног очувања запремине и јасно дефинисане границе између флуида и ваздуха, омогућавајући убедљиве визуелне ефекте чак и при коришћењу упрошћених физичких модела. LSM, са својом имплицитном репрезентацијом граничне површине, погодан је за симулацију сложених тополошких промена, као што су спајање и раздвајање течних маса, прскање и формирања пене. Овај приступ омогућава прецизно израчунавање кривине и нормале на површини, чиме се омогућава тачно моделовање површинског напона и капиларности, што побољшава визуелну веродостојност. PFM се ретко користи у комерцијалним алатима за визуелне ефекте због високе рачунске захтевности и фокуса на физичку прецизност. Ипак, њен теоријски оквир значајно утиче на развој напредних модела у научним симулацијама двофазних токова.

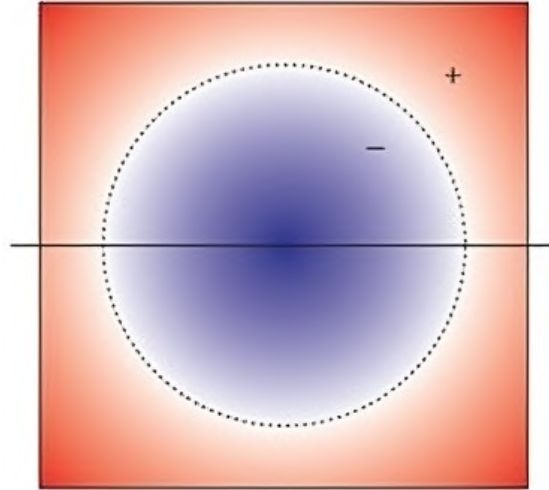
У овом раду користи се LSM метода, при чему се кретање границе одвија под утицајем спољашњег векторског поља брзина $\mathbf{u}(\mathbf{x}, t)$, добијеног нумеричким решавањем Навије-Стоксових једначина. За њену нумеричку имплементацију користи се једна додатна метода, такође заснована на имплицитним функцијама. Детаљнији опис обе методе дат је у наредном одељку.

4.3 Методе засноване на имплицитним функцијама

Ова група метода заснива се на представљању границе Γ домена $\Omega \subset \mathbb{R}^d$ преко неке имплицитно задате функције $\phi(\mathbf{x})$. Ако користимо функцију означеног растојања, функција ϕ се дефинише као [13]:

$$\phi(\mathbf{x}) = \begin{cases} \phi(\mathbf{x}) < 0, & \text{ако } \mathbf{x} \in \Omega, \\ \phi(\mathbf{x}) = 0, & \text{ако } \mathbf{x} \in \Gamma := \partial\Omega, \\ \phi(\mathbf{x}) > 0, & \text{ако } \mathbf{x} \in \bar{\Omega}. \end{cases} \quad (4.1)$$

Функција ϕ представља растојање до границе Γ , тако да је $\phi > 0$ на једној страни, а $\phi < 0$ на другој страни границе. Када се гранична површина креће, постаје зависна од времена t , па је у сваком тренутку дефинисана као изоповршина $\phi(\mathbf{x}, t) = 0$. Тада важи $|\nabla\phi| = 1$, односно да је норма градијента функције ϕ једнака јединици. Уколико је Γ затворена површина, ϕ је позитивна ван, а негативна унутар Γ . На Слици 4.1 приказана је функција означеног растојања у случају када је Γ круг.



Слика 4.1: Функција означеног растојања за круг [13]. Позитивне вредности функције приказане су црвеном бојом, а негативне плавом. Областима јачег интензитета боја одговарају веће апсолутне вредности функције растојања.

Функција ϕ не мора да буде функција растојања, али нумеричке апроксимације губе на тачности ако градијент функције значајно одступа од јединице

— односно ако $|\nabla\phi|$ знатно варира. Геометријске карактеристике граничне површине се могу израчунати директно из ϕ . Јединични вектор нормале $\mathbf{n}$ и средња кривина κ су дати изразима:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad \text{и} \quad \kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}. \quad (4.2)$$

Да би се померила гранична површина, неопходно је дефинисати једначину еволуције за целокупну функцију ϕ . Нека је $\mathbf{x} \in \Gamma$ тачка на граничној површини. У случају покретне граничне површине, свакој тачки $\mathbf{x}$ на њој придружена је брзина кретања описана вектором $\mathbf{u}$, који зависи од конкретне примене. У нашем случају, он је одређен струјањем флуида, а у другим применама може зависити од локалне геометрије граничне површине или других фактора.

Вектор брзине $\mathbf{u}$ се може разложити на своју нормалну и тангентну компоненту: $\mathbf{u} = \mathbf{u}^\perp + \mathbf{u}^\parallel$. Ако је $\mathbf{u}^\perp = 0$ за сваку тачку $\mathbf{x} \in \Gamma$, тада ће гранична површина остати непомицна, јер је целокупан ток тангентан на њу. Једино нормална компонента брзине $\mathbf{u}^\perp$ доприноси кретању граничне површине. Нека за $\mathbf{u}^\perp$ важи $\mathbf{u}^\perp(\mathbf{x}, t) = F(\mathbf{x}, t)\mathbf{n}$. Тада се кретање тачке $\mathbf{x}$ може описати помоћу једначине:

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}, t)\mathbf{n}. \quad (4.3)$$

Ово је основна једначина кретања тачке која припада граничној површини. У пракси, израчунавање функције F зависи од конкретног проблема. Постоје две основне методе које се могу користити за праћење еволуције овако дефинисане граничне површине. У наставку ћемо их детаљније описати.

Метода брзог марширања

У *методи брзог марширања* (енг. *Fast Marching Method* - FMM) кретање граничне површине се описује помоћу функције познате као *маја времена преласка* (енг. *time of crossing map*), где вредност $\phi(\mathbf{x})$ представља тренутак када гранична површина први пут пролази кроз тачку $\mathbf{x}$. Према томе, гранична површина у тренутку t се дефинише као изоповршина [12]:

$$\phi(\mathbf{x}(t)) = t. \quad (4.4)$$

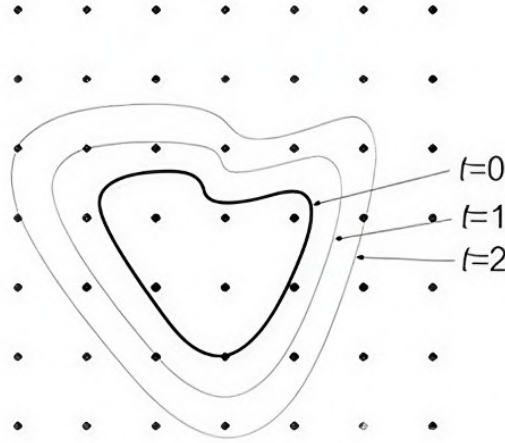
Диференцирањем једначине (4.4) добија се:

$$\frac{d\phi(\mathbf{x}(t))}{dt} = \nabla\phi \cdot \frac{\partial\mathbf{x}}{\partial t} = \nabla\phi \cdot \mathbf{u} = 1, \quad (4.5)$$

која уз (4.3) даје једначину:

$$F(\mathbf{x})|\nabla\phi| = 1, \quad (4.6)$$

познату као *једначина ејконала* која описује проблем са граничним условима. На Слици 4.2 приказано је кретање контуре дефинисане једначином (4.4).



Слика 4.2: Еволуција контуре у равни, дефинисане једначином (4.4), за различите временске тренутке [12].

Метода брзог марширања има и значајна ограничења, јер не дозвољава да међуповршина прелази преко својих претходних позиција током времена. Кретање мора бити строго монотono у односу на почетни положај, или ка унутрашњости или ван ње. Детаљи о нумеричком поступку којим се једначина (4.6) може решити могу се наћи у [12]. Ова метода се обично користи у оквиру једног корака нумеричке имплементације методе ниво скупа.

Метода ниво скупа

У методи ниво скупа гранична површина се слободно развија током времена, без ограничења на тополошке промене. Овај приступ може моделовати кретања која се одвијају у оба смера, као и прелазе преко већ обухваћених области, што није дозвољено у FMM. Кретање граничне површине се моделује тако што се укупни извод функције $\phi(\mathbf{x}(t), t)$ по времену изједначи са нулом [13]:

$$\frac{d}{dt}\phi(\mathbf{x}(t), t) = 0 \quad \Rightarrow \quad \frac{\partial\phi}{\partial t} + \frac{d\mathbf{x}}{dt}\nabla\phi = 0, \quad (4.7)$$

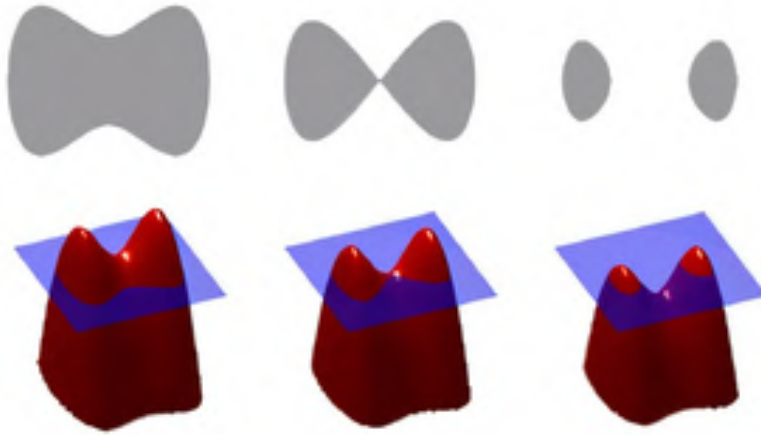
односно

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \nabla \phi = 0. \quad (4.8)$$

Узимајући у обзир једначине (4.2) и (4.3), једначина (4.8) се може записати у облику:

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0. \quad (4.9)$$

Ово је основна једначина еволуције граничне површине и припада, под одређеним условима, класи Хамилтон-Јакобијевих једначина. Једначине (4.8) и (4.9) се користе равноправно у случају споља задатих поља брзине, као што су проблеми у којима је брзина одређена кретањем флуида. Међутим, када су поља брзине унутрашња и самогенерисана (на пример, када зависе од локалне кривине граничне површине), мора се применити модификовани приступ заснован на једначини (4.9), како би се тачно моделовала динамика граничне површине.



Слика 4.3: Промена топологије контуре у равни [13]. Црвеном бојом су представљене вредности функције ϕ , док плаве пресечне равни одговарају различитим временским тренуцима.

Репрезентација преко функција ниво скупа има бројне теоријске и практичне предности. Имплицитна дефиниција површине као нултог ниво скупа омогућава велики број степени слободе, тополошку флексибилност и прецизно моделовање сложених, неправилних облика. На Слици 4.3 илустрована је промена топологије контуре у равни. У пракси, једино ограничење представљања међуповршине преко ниво скупа поставља резолуција рачунске мреже, која директно утиче на тачност и глаткоћу добијене граничне површине. Ова

комбинација флексибилности, ефикасности и математичке елеганције чини методе засноване на ниво скуповима моћним алатом у рачунарској графичи.

Нумеричко решавање једначине кретања граничне површине

Еволуција граничне површине, представљене функцијом $\phi(\mathbf{x}, t)$, добија се нумеричким решавањем једначина (4.8) или (4.9). Функција ϕ је дефинисана на неком домену на коме је дефинисан почетни положај граничне површине. Да би се моделовала функција ϕ , неопходно је познавати одговарајуће почетне услове. У динамичкој формулацији, за разлику од статичких метода, функција F мора бити дефинисана не само на самој граничној површини, већ на целом домену где је дефинисана и функција ϕ . Стога је неопходно проширити поље брзине на цео тај домен.

Дакле, први корак у моделовању кретања граничне површине је иницијализација функције ϕ . Уколико је почетна површина задата експлицитно, или параметарски, то може бити нетривијалан проблем. Међутим, гранична површина се најчешће задаје у виду комбинације стандардних геометријских облика или као површина представљена мрежом троуглова у неком од уобичајених формата. Ово значајно поједностављује поступак, с обзиром на доступност великог броја софтверских алата помоћу којих се почетна гранична површина може добити у облику нулте изоповршине.

Следећи корак је дискретизација једначине (4.9), односно једначине (4.8), која се обично користи у анализи кретања флуида. За просторну дискретизацију ове једначине у пракси се најчешће користи метод коначних разлика, што омогућава једноставно спајање са прорачунима флуидних токова коришћењем исте рачунске мреже. Постоје приступи који користе посебну, финију мрежу за међуповршину ради побољшања резолуције, нарочито на слободној површини флуида [38]. Ипак, стандардна пракса подразумева коришћење заједничке мреже како за решавање Навије-Стоксових једначина, тако и за решавање једначина кретања граничне површине. Будући да је једначина (4.8) по својој природи адвекциона, за њено решавање може се применити иста методологија која се користи за Навије-Стоксове једначине.

У контексту двофазних токова, решавање једначина еволуције ниво скупова најчешће се реализује семи-Лагранжовом методом, с обзиром на то да

се она већ користи у оквиру решавања Навије-Стоксових једначина. У односу на експлицитну Ојлерову методу комбиновану са *upwind* (или WENO) диференцијском шемом за просторну дискретизацију, овај приступ обезбеђује већу нумеричку стабилност и тачност, чак и при већим временским корацима. По једноставности имплементације он надмашује имплицитне методе, јер не захтева решавање линеарних система једначина. Међутим, за спречавање нумеричких артефаката, неопходна је квалитетна интерполациона шема, најчешће кубна или сплајн интерполација. Због доброг баланса између стабилности, ефикасности и квалитета резултата, семи-Лагранжова метода је посебно погодна за симулације флуида у рачунарској графици, где је циљ постићи визуелно убедљиве, али и рачунски ефикасне симулације.

При решавању једначине (4.9) претпоставља се да је брзина $F(\mathbf{x})$ позната у целом домену. Ово је задовољено када су обе фазе течности, али ситуација је знатно компликованија када је једна од фаза ваздух, односно када се анализира ток течности у ваздуху. Са F_{ext} означимо проширење функције F на цео домен. Једначина из које се може одредити функција F_{ext} добија се из једначине (4.9) и услова $|\nabla\phi| = 1$:

$$\nabla F_{\text{ext}} \nabla \phi_{\text{tren}} = 0, \quad (4.10)$$

где је ϕ_{tren} вредност функције ϕ добијена у последњем нумеричком кораку. Горњу једначину је потребно решити за F_{ext} само у области испуњеној ваздухом. Постоји неколико начина за решавање ове једначине. Први од њих подразумева решавање једначине (4.10) већ поменутом FMM методом, с обзиром да има исту форму као (4.6). Показало се, међутим, да је једноставније конвертовати је у временски зависан проблем:

$$\frac{\partial F_{\text{ext}}}{\partial t} + \text{sign}(\phi_{\text{tren}}) \frac{\nabla \phi_{\text{tren}}}{|\nabla \phi_{\text{tren}}|} \cdot \nabla F_{\text{ext}} = 0. \quad (4.11)$$

Горња једначина је хиперболичка једначина чије карактеристичне криве су усмерене од граничне површине, у правцу нормале на њу. У софтверским пакетима намењеним LSM, обично су имплементиране обе методе.

Након адвекционог корака, функција ϕ губи својство функције означеног растојања ($|\nabla\phi| \neq 1$), што директно угрожава тачност прорачуна нормале и кривине. Како знак функције и даље остаје исправан, неопходно је извршити њену периодичну реиницијализацију (сваких 5 до 10 временских корака) ре-

шавањем једначине:

$$\frac{\partial \phi}{\partial \tau} = \text{sign}(\phi_{\text{tren}}) (1 - |\nabla \phi|), \quad \text{уз почетни услов } \phi(0) = \phi_{\text{tren}}, \quad (4.12)$$

где је τ псеудо-временски параметар који нема физичко значење, већ служи искључиво за итеративно приближавање стационарном решењу једначине $|\nabla \phi| = 1$. У сложенијим симулацијама, функција ϕ се израчунава у уском појасу око границе (тзв. *narrow band*), како би се минимизирали рачунски захтеви. При томе, одржава се висока тачност у праћењу слободне површине, чак и у комплексним токовима са великим деформацијама и тополошким променама. И реиницијализација се, наравно, врши искључиво у том уском појасу.

4.4 Флуиди са слободном површином

Флуиди са слободном површином (енг. *free-surface flow*) представљају посебну класу двофазних струјања флуида. Њихово моделовање укључује прорачун кретања флуида са јасно дефинисаном границом између две фазе, на пример воде и ваздуха, где је друга фаза знатно ређа. Та граница се слободно креће и деформише под утицајем струјања, гравитације, површинског напона и спољашњих сила. Осим у рачунарској графици, овај приступ има широку примену у инжењерским анализама. Користи се за симулацију таласа, преливања преко брана, кретања течности у делимично напуњеним резервоарима, потапања или испливавања објеката, као и формирања мехурова.

Главни изазов у моделовању овог типа флуида лежи у тачном праћењу и описивању променљиве динамике слободне површине. Струјање флуида у обе фазе одређује се одговарајућим Навије-Стоксовим једначинама и граничним условима који важе на површини која их раздваја. Навије-Стоксове једначине написаћемо у стандардном облику [36]:

$$\frac{D\mathbf{u}_\ell}{Dt} = -\frac{1}{\rho_\ell} \nabla p_\ell + \frac{\mu_\ell}{\rho_\ell} \Delta \mathbf{u}_\ell + \mathbf{g}, \quad \nabla \cdot \mathbf{u}_\ell = 0, \quad \mathbf{x} \in \text{течности} \quad (4.13)$$

$$\frac{D\mathbf{u}_g}{Dt} = -\frac{1}{\rho_g} \nabla p_g + \frac{\mu_g}{\rho_g} \Delta \mathbf{u}_g + \mathbf{g}, \quad \nabla \cdot \mathbf{u}_g = 0, \quad \mathbf{x} \in \text{гасу} \quad (4.14)$$

где је $\mathbf{u}$ брзина, p притисак, ρ густина, а μ (динамичка) вискозност флуида. Индекси ℓ и g означавају течну и гасовиту фазу редом, D/Dt је материјални извод, а $\mathbf{g}$ убрзање услед гравитације.

На површини између две фазе важе *кинематички* и *динамички гранични услови*. Кинематички услов намеће да честице флуида не могу пролазити кроз границу, односно да се граница креће заједно са флуидом. Овај услов се математички формулише као:

$$\mathbf{u}_l \cdot \mathbf{n} = \mathbf{u}_g \cdot \mathbf{n}, \quad (4.15)$$

тј. нормалне компоненте брзине обе фазе морају бити једнаке. Динамички услов изражава континуитет тангенцијалних компоненти напона на граничној површини, односно обезбеђује да разлика у напонима између две фазе буде уравнотежена површинским напонам. Овај услов је дат релацијом [36]:

$$[-p_l + \mu_l \mathbf{n}(\nabla \mathbf{u}_l + (\nabla \mathbf{u}_l)^T) \mathbf{n}] - [-p_g + \mu_g \mathbf{n}(\nabla \mathbf{u}_g + (\nabla \mathbf{u}_g)^T) \mathbf{n}] = \gamma \kappa, \quad \mathbf{x} \in \Gamma, \quad (4.16)$$

где је $\mathbf{n}$ јединична нормала на граничну површину усмерена из гаса ка течности, γ површински напон, а κ средња кривина граничне површине повезане са функцијом ϕ релацијом (4.2). Може се уочити да због дејства површинског напона γ , долази до скока притиска p , нормално на граничну површину, који је пропорционалан средњој кривини површине κ . У нашим прорачунима ћемо занемарити површински напон. С обзиром на то да анализирамо само невискозне флуиде, услов (4.16) се своди на: $p_l = p_g = 0$.

Једначине Навије-Стокса за двофазно струјање флуида (4.13) и (4.14) су спрегнуте са једначином (4.8) која описује еволуцију њихове границе. Дакле, алгоритму описаном на крају одељка 3.3 неопходно је додати корак за израчунавање нове вредности функције ϕ , чиме се добија проширени облик алгоритма. За сваки временски корак, извршавају се следећи кораци:

- Адвекција $\mathbf{u}^n$: $\mathbf{u}^A = \text{advect}(\mathbf{u}^n, \Delta t, \mathbf{u}^n)$;
- Урачунавање деловања запреминских сила: $\mathbf{u}^B = \mathbf{u}^A + \Delta t \mathbf{g}$;
- Пројекција притиска: $\mathbf{u}^{n+1} = \text{project}(\Delta t, \mathbf{u}^B)$;
- Адвекција ϕ^n : $\phi^{n+1} = \text{advect}(\phi^n, \Delta t, \mathbf{u}^n)$.

Глава 5

Ковекторски приступ анализи кретања флуида

5.1 Историјски корени методе

Током година, формулације засноване на ковекторима више пута су поново увођене у динамику флуида. Ковектори и диференцијалне форме коришћени су још у XVIII веку у раним проучавањима нестишљивих флуида заснованих на Ојлеровим једначинама [39]. Овај формализам довео је до открића важних закона одржања у оквиру Лагранжовог приступа, укључујући одржање вртлога, ковекторских поља брзине и њихових линијских интеграла дуж затворених и отворених кривих.

Средином XIX века интересовање за формализам заснован на ковекторима је опало, што је довело до привременог застоја у његовој примени. Међутим, тај приступ се поново појавио у Арнолдовој геометријској хидродинамици, где је уведен револуционарни геометријски поглед на кретање флуида [40]. Уместо да се флуид третира као скуп честица под дејством сила, он се посматра као бесконачно димензионални динамички систем, чији конфигурациони простор чини група дифеоморфизама који очувавају запремину. Кретање флуида тада постаје геодезијско у односу на метрику дефинисану кинетичком енергијом, а Ојлерове једначине за нестишљив флуид добијају се као геодезијске једначине на тој групи. Касније су Марсден и Вајнштајн проширили ову визију коришћењем метода савремене геометријске механике [41]. Ковекторски формализам је уведен у Хамилтонову формулацију динамике флуида, а у савременој теорији нашао је примену у развоју такозваних им-

пулсних метода [42]. О односу импулсних метода и ковекторског приступа представљеног у овом раду биће речи нешто касније у овој глави.

Ковекторски формализам (детаљно представљен у [11, 43]) који чини идејну основу овог мастер рада, не наглашава експлицитно повезаност са теоријом симетрија и геометријском механиком. Међутим, увођење 1-форми као основних променљивих за опис брзине флуида директно произилази из Арнолдовог геометријског приступа. Тиме се обезбеђује да симулација поштује кључне физичке принципе, попут очувања вртложности, чиме се овај приступ поставља као дискретна реализација геометријске теорије динамике флуида.

Упркос порасту броја радова о моделовању флуида ковекторским приступом, имплементација ковекторских структура у нумеричке моделе динамике флуида и даље остаје недовољно истражена. Овим се отвара простор за развој геометријски заснованих нумеричких шема с циљем постизања вишег реда тачности и побољшане стабилности. Такви приступи имају потенцијал да значајно унапреде ефикасност и прецизност симулација динамике флуида.

5.2 Основни појмови теорије диференцијалних форми

У даљем тексту биће изложени основни елементи теорије диференцијалних форми неопходни за разумевање ковекторског приступа динамици флуида [44, 45]. На први поглед, овај приступ се може чинити значајно различитим у односу на традиционалну методологију засновану на векторској анализи, која је описана у претходним поглављима. С обзиром на апликативну природу овог рада, излагање ће бити у великој мери интуитивно, са циљем да омогући приступачно и јасно разумевање кључних концепата.

Постоје два основна разлога за увођење диференцијалних форми у опис физичких поља. Први потиче из чињенице да физичка мерења увек дају глобалне величине, тј. вредности придружене оријентисаним геометријским елементима као што су криве, површине и запремине. Ове глобалне величине заправо представљају интеграле локалних вредности поља по поменутиим геометријским елементима. Уопштено говорећи, сматра се да су физички релевантне величине искључиво оне доступне експерименталном одређивању.

Други разлог проистиче из чињенице да интегралне формулације закона очувања, изражене језиком диференцијалних форми, омогућавају развој ефикаснијих и робустнијих нумеричких метода у односу на приступе засноване на диференцијалним једначинама. Такве формулације обезбеђују аутоматско важење закона очувања и у дискретним репрезентацијама, што није увек гарантовано при дискретизацији диференцијалних једначина. Посебна пажња ће бити посвећена Лијевом изводу диференцијалних форми, који чини основу за имплементацију новог адвекционог корака у оквиру раније описаног алгорита за нумеричко решавање Ојлерових једначина. Лијев извод обезбеђује одржање вртложности током симулације флуида, чиме се постиже виши степен физичке верности и нумеричке стабилности.

Многострукости, тангентни и котангентни простори

Практично коришћење концепта многострукости започето је много пре његове формалне математичке дефиниције. На пример, у средњем веку морепловци су кретање бродова приказивали на географским картама, које представљају дводимензионалне пројекције делова Земљине површине, а не као померање у тродимензионалном простору $\mathbb{R}^3$. Још изразитији пример је кретање возова и возила по железничким пругама и путевима. С обзиром на то да се целокупна Земљина површина не може довољно детаљно приказати на једној карти, коришћене су колекције карти, односно атласи. Ови појмови, карте и атласи, задржани су и у диференцијалној геометрији.

Интуитивно, многострукост је тополошки простор који се локално посматрано „понаша” као еуклидски простор (нпр. $\mathbb{R}^n$), али глобално може бити сложен и закривљен. Прецизније, тополошки простор M је многострукост димензије n ако је Хаусдорфов простор, ако поседује пребројиву базу (секундарно је пребројив) и ако за сваку тачку $p \in M$ постоји отворена околина $U \subseteq M$ и хомеоморфизам $\phi : U \rightarrow \tilde{U}$, где је $\tilde{U}$ отворен скуп у $\mathbb{R}^n$.

Уређени пар (U, ϕ) назива се *локална карта* (енг. *chart*), а пресликавање ϕ координатна функција (нпр. $\phi(p) = (x^1, x^2, \dots, x^n)$). Скуп локалних карата $\{(U_\alpha, \phi_\alpha)\}_{\alpha \in A}$ такав да је $\bigcup_{\alpha \in A} U_\alpha = M$ назива се *атлас*. Атлас се назива глатким ако за било које две карте (U, ϕ) и (V, ψ) чији је пресек непразан скуп ($U \cap V \neq \emptyset$), пресликавање $\psi \circ \phi^{-1} : \phi(U \cap V) \rightarrow \psi(U \cap V)$ представља дифеоморфизам (глатко пресликавање са глатким инверзом).

Многострукост на којој је дефинисан глатки атлас назива се *глатка многострукост*. Многострукости омогућавају да се физички закони формулишу на начин који је независан од избора координатног система и има геометријски смисао чак и када се ради на закривљеним или дискретизованим површинама.

На свакој тачки $p \in M$ можемо дефинисати *тангентни простор* $T_p M$, који представља векторски простор димензије n који садржи све могуће смерове кретања кроз ту тачку. Формално, елементи простора $T_p M$ могу бити дефинисани као класе еквиваленције глатких кривих $\gamma(t)$ у M које пролазе кроз p (таквих да је $\gamma(0) = p$), где вектор представља брзину промене $\dot{\gamma}(0)$.

У локалним координатама $(x^1, \dots, x^n)$ око тачке p , сваки тангентни вектор $\mathbf{v} \in T_p M$ се може записати као линеарна комбинација:

$$\mathbf{v} = v^i \frac{\partial}{\partial x^i} \Big|_p, \quad (5.1)$$

где индекси који се понављају подразумевају сумирање (Ајнштајнова конвенција). Симбол $\frac{\partial}{\partial x^i}$ је, у овом контексту, стандардна ознака за координатни базни вектор тангентног простора у тачки p (односно за координатно векторско поље). Скуп $\{\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^n}\}$ чини базу тангентног простора $T_p M$. Пресликавање које свакој тачки $p \in M$ придружује један вектор из $T_p M$ назива се векторско поље дефинисано на M .

Дуалан овом простору је *котангентни простор* $T_p^* M$, који се састоји од свих линеарних пресликавања $\alpha : T_p M \rightarrow \mathbb{R}$. Елементи овог простора називају се *ковектори* (или линеарне 1-форме). У истим локалним координатама, дуална база котангентног простора је дата диференцијалима координата $\{dx^1, \dots, dx^n\}$, тако да важи релација дуалности:

$$dx^i \left(\frac{\partial}{\partial x^j} \right) = \delta_j^i, \quad (5.2)$$

где је δ_j^i Кронекеров делта симбол. Стога, сваки ковектор се може записати као:

$$\alpha = \alpha_i dx^i, \quad (5.3)$$

где су α_i његове компоненте. Овај запис јасно разликује доње индексе (компоненте ковектора) од горњих (компоненте вектора). *Природно сјаривање* ковектора α и вектора $\mathbf{v}$ дато је са:

$$\alpha(\mathbf{v}) = \alpha_i v^i \in \mathbb{R} \quad (5.4)$$

и не захтева дефинисање метрике. Унија свих котангентних простора за сваку тачку многострукости:

$$T^*M = \bigcup_{p \in M} T_p^*M, \quad (5.5)$$

назива се *котангентно раслојење* (енг. *cotangent bundle*) многострукости M .

Ковекторско поље се дефинише као глатко пресликавање које свакој тачки p многострукости M додељује ковектор $\alpha_p \in T_p^*M$ и назива се диференцијална 1-форма. У општем случају, диференцијална k -форма је глатко поље које свакој тачки p многострукости M додељује антисиметричну линеарну k -форму:

$$\omega : \underbrace{T_p M \times \cdots \times T_p M}_k \rightarrow \mathbb{R}. \quad (5.6)$$

Практично, диференцијалне форме су природни објекти за геометријску интеграцију:

- 0-форме одговарају локалним скаларним пољима (нпр. притисак, температура);
- 1-форме одговарају векторским пољима која се интегришу по кривама (нпр. рад силе дуж путање);
- 2-форме одговарају векторским пољима која се интегришу по површинама (нпр. флуks поља кроз површину);
- 3-форме одговарају скаларним пољима која се интегришу по запреминама (нпр. густина масе или наелектрисања).

Интеграли диференцијалних форми по оријентисаним геометријским елементима сматрају се апстракцијом процеса мерења, јер омогућавају дефинисање глобалних физичких величина, доступних експерименталном одређивању.

Када су у питању примене у механици флуида (и физици генерално), M обично представља просторно-временску област, па су онда:

- $T_p M$ — могући правци (брзине) кретања честице у p ,
- $T_p^* M$ — могући импулси у тачки p (импулс је природно ковектор),
- $T^* M$ — фазни простор (положај + импулс).

Спољашњи производ и спољашњи извод

За диференцијални и интегрални рачун на многострукостима користи се термин *спољашњи рачун* (енг. *exterior calculus*). У спољашњем рачуну за диференцијалне форме се дефинишу два фундаментална оператора: *спољашњи*

производ и спољашњи извод. Они омогућавају јединствену и елегантну формулацију диференцијалних операција и интегралних теорема. Ови алати уопштавају познате појмове из векторске анализе на начин који је геометријски интуитиван, координатно независан и кога је лако уопштити на произвољне димензије.

Спољашњи производ (енг. *wedge product*, $\wedge$) је билинеарна, антисиметрична операција којом се две диференцијалне форме комбинују у форму вишег реда. За 1-форме α_1 и α_2 спољашњи производ $\alpha_1 \wedge \alpha_2$ је 2-форма дата са:

$$\alpha_1 \wedge \alpha_2(\mathbf{v}_1, \mathbf{v}_2) := \det[\alpha_i(\mathbf{v}_j)]_{i,j=1,2}. \quad (5.7)$$

Из антисиметричности детерминанте следи да је $\alpha_1 \wedge \alpha_2$ антисиметрично преликавање. Такође, важи: $\alpha_1 \wedge \alpha_2 = -\alpha_2 \wedge \alpha_1$ и $\alpha \wedge \alpha = 0$. Геометријски, $\alpha_1 \wedge \alpha_2$ представља оријентисану површину (елемент 2-форме) која одговара паралелограму дефинисаном векторима на којима се мере α_1 и α_2 . На пример, у $\mathbb{R}^3$, производ $dx \wedge dy$ одговара оријентисаном елементу површине у x - y равни. Производ три 1-форме, као што је $dx \wedge dy \wedge dz$, даје оријентисани запремински елемент, основни елемент за интеграцију по запремини. Спољашњи производ је асоцијативан и задовољава Лајбницово правило у комбинацији са спољашњим изводом.

Спољашњи извод (енг. *exterior derivative*, d) је линеарни оператор који преликава k -форму у $(k+1)$ -форму. Његова дефиниција је чисто алгебарска и зависи само од глаткости поља, а не од метрике или координата. Његово најважније својство је:

$$d^2 = 0, \quad \text{тј.} \quad d(d\omega) = 0, \quad (5.8)$$

познато као Поенкареова лема, која одговара фундаменталним идентитетима у векторској анализи.

Спољашњи извод уопштава три основне операције:

- За 0-форму (скаларно поље f):

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial z} dz. \quad (5.9)$$

Ово је диференцијал функције тј. 1-форма која одговара градијенту. У векторској анализи, ово је вектор ∇f , али у спољашњем рачуну, градијент је природно ковектор (1-форма).

- За 1-форму $\alpha = \alpha_x dx + \alpha_y dy + \alpha_z dz$:

$$d\alpha = \left(\frac{\partial \alpha_y}{\partial x} - \frac{\partial \alpha_x}{\partial y} \right) dx \wedge dy + \left(\frac{\partial \alpha_z}{\partial y} - \frac{\partial \alpha_y}{\partial z} \right) dy \wedge dz + \left(\frac{\partial \alpha_x}{\partial z} - \frac{\partial \alpha_z}{\partial x} \right) dz \wedge dx. \quad (5.10)$$

Ово је 2-форма која одговара ротору векторског поља, тј. ако 1-форма α одговара векторском пољу $\mathbf{v}$, тада је $d\alpha$ 2-форма која одговара вектору $\nabla \times \mathbf{v}$.

- За 2-форму $\beta = \beta_{xy} dx \wedge dy + \beta_{yz} dy \wedge dz + \beta_{zx} dz \wedge dx$:

$$d\beta = \left(\frac{\partial \beta_{xy}}{\partial z} + \frac{\partial \beta_{yz}}{\partial x} + \frac{\partial \beta_{zx}}{\partial y} \right) dx \wedge dy \wedge dz. \quad (5.11)$$

Ово је 3-форма која одговара дивергенцији. Ако 2-форма β представља флукс векторског поља $\mathbf{F}$ кроз површину, тада њен спољашњи извод представља меру локалног настајања или нестајања, тј. $\nabla \cdot \mathbf{F}$.

Спољашњи извод и теорема Стокса су у основи интегралног рачуна на многострукостима. Општа теорема Стокса гласи:

$$\int_{\Omega} d\omega = \int_{\partial\Omega} \omega, \quad (5.12)$$

где је Ω оријентисана $(k+1)$ -димензионална подмногострукост са границом $\partial\Omega$, док ω представља k -форму. Ова релација уопштава класичне теореме анализе:

- Њутн-Лајбницову теорему ($k=1$):

$$\int_a^b df = f(b) - f(a). \quad (5.13)$$

- Теорему Грина (класична теорема Стокса) ($k=2$):

$$\int_S (\nabla \times \mathbf{F}) \cdot d\mathbf{S} = \oint_{\partial S} \mathbf{F} \cdot d\mathbf{r}. \quad (5.14)$$

- Теорему дивергенције (Гаусова теорема) ($k=3$):

$$\int_V \nabla \cdot \mathbf{F} dV = \oint_{\partial V} \mathbf{F} \cdot d\mathbf{S}. \quad (5.15)$$

Ове теореме показују да је локална промена, описана са $d\omega$, у вези са глобалним понашањем на границама.

Музички изоморфизми и диференцијал функције

Музички изоморфизми (енг. *Musical isomorphisms*) су линеарна пресликавања између тангентног и котангентног простора, дефинисана у складу са релацијом (5.4). Напоменимо да дефинисање скаларног производа подразумева претходно увођење метрике у многострукост. По дефиницији многострукост је тополошки простор и не мора имати дефинисану метрику. Музички изоморфизми се заснивају на скаларном производу, па према томе претпостављају да је метрика дефинисана.

Снизилица (енг. *flat*), у ознаци $\flat$, пресликава вектор у ковектор. За вектор $\mathbf{v} \in T_p M$, ако је метрика дата тензором g_{ij} , одговарајући ковектор је:

$$\mathbf{v}^\flat = g_{ij} v^j dx^i. \quad (5.16)$$

Повисилица (енг. *sharp*), у ознаци $\sharp$, пресликава ковектор у вектор. За ковектор $\alpha \in T_p^* M$, одговарајући вектор $\alpha^\sharp$ је дефинисан са:

$$\alpha^\sharp = g^{ij} \alpha_j \frac{\partial}{\partial x^i}, \quad (5.17)$$

где је g^{ij} инверзна матрица метричког тензора ($g^{ik} g_{kj} = \delta_j^i$). Музички изоморфизми омогућавају прелаз између вектора и ковектора, што је неопходно када желимо да повежемо геометријске појмове (као што је брзина као 1-форма) са физичким интерпретацијама у векторском облику. За еукледску метрику у $\mathbb{R}^3$ важи $g_{ij} = \delta_j^i$, па се компоненте вектора и ковектора поклапају, што је разлог зашто се у класичној механици флуида ова разлика често имплицитно занемарује.

Ако је $f : M \rightarrow \mathbb{R}$ глатка функција дефинисана на многострукости M , њен диференцијал $df(p)$ је диференцијална 1-форма дефинисана тако да за сваки тангентни вектор $\mathbf{v} \in T_p M$, важи:

$$df(\mathbf{v}) = \partial_{\mathbf{v}} f, \quad (5.18)$$

где $\partial_{\mathbf{v}} f$ означава извод функције f у правцу вектора $\mathbf{v}$. Ово значи да df није вектор, већ ковектор - елемент котангентног простора $T_p^* M$.

Оператори повлачења и гурања

Нека је дато глатко пресликавање $\Psi : M \rightarrow N$ између две глатке многострукости:

$$\Psi(x_1, \dots, x_n) = (\Psi^1(x_1, \dots, x_n), \Psi^2(x_1, \dots, x_n), \dots, \Psi^n(x_1, \dots, x_n)), \quad (5.19)$$

где су $\{x_i\}$ локалне координате тачке p из M . Дефинишу се *оператор гурања* (енг. *pushforward*), у ознаци Ψ_* , и *оператор повлачења* (енг. *pullback*), у ознаци Ψ^* , који омогућавају трансформацију тангентних и котангентних објеката између простора M и N .

Гурање делује на тангентне векторе: за сваку тачку $p \in M$, дефинише се линеарно пресликавање $\Psi_* : T_p M \rightarrow T_{\Psi(p)} N$, које „гура напред” вектор из $T_p M$ коришћењем матрице $\mathbf{J}_\Psi(\mathbf{x})$ чији су елементи $(\mathbf{J}_\Psi)_i^j = \partial \Psi^j / \partial x^i$. Матрица $\mathbf{J}_\Psi(\mathbf{x})$ је Јакобијева матрица пресликавања Ψ и реализује његову линеаризацију:

$$\Psi(\mathbf{x} + \Delta \mathbf{x}) = \Psi(\mathbf{x}) + \mathbf{J}_\Psi(\mathbf{x}) \cdot \Delta \mathbf{x}. \quad (5.20)$$

Ако су $\{x_i\}$ локалне координате тачке p из M , а $\{y_j\}$ локалне координате тачке $\Psi(p)$ у N , тада је деловање Ψ_* на базни вектор $\partial / \partial x_i$ дато са:

$$\Psi_* \left(\frac{\partial}{\partial x^i} \right) = \sum_j \frac{\partial \Psi^j}{\partial x^i} \frac{\partial}{\partial y^j}, \quad (5.21)$$

где су Ψ^j компоненте пресликавања Ψ . Онда је:

$$\Psi_*(\mathbf{v}) = \mathbf{J}_\Psi(p) \mathbf{v}, \quad \forall \mathbf{v} \in T_p M. \quad (5.22)$$

С друге стране, повлачење Ψ^* делује на диференцијалне форме: $\Psi^* : T_{\Psi(p)}^* N \rightarrow T_p^* M$. Повлачење $\Psi^* \omega$ диференцијалне 1-forme ω на N дефинише се као форма на M тако да важи:

$$\Psi^* \omega(\mathbf{v}) = \omega(\Psi_*(\mathbf{v})), \quad \forall \mathbf{v} \in T_p M, \quad (5.23)$$

што осигурава одржавање дуалности између вектора и ковектора. Кључно својство оператора повлачења је да комутира са спољашњим изводом:

$$\Psi^*(d\omega) = d(\Psi^* \omega), \quad (5.24)$$

док за интеграцију важи:

$$\int_C \Psi^* \omega = \int_{\Psi(C)} \omega, \quad (5.25)$$

у случају када је Ψ дифеоморфизам, што је изузетно важно за дискретизацију закона очувања.

У координатном запису, ако је $\omega = \sum_j \alpha_j(y) dy^j$, повлачење се израчунава заменом y^j са $\Psi^j(x)$ и применом правила диференцирања сложених функција:

$$\Psi^* \omega = \sum_j \alpha_j(\Psi(x)) d(\Psi^j) = \sum_j \alpha_j(\Psi(x)) \left(\sum_i \frac{\partial \Psi^j}{\partial x^i} dx^i \right) = \sum_i \left(\sum_j \frac{\partial \Psi^j}{\partial x^i} \alpha_j(\Psi(x)) \right) dx^i. \quad (5.26)$$

Еквивалентни матрични запис ове релације има облик:

$$(\Psi^* \omega)(\mathbf{x}) = (\mathbf{J}_\Psi(\mathbf{x}))^T \omega(\Psi(\mathbf{x})). \quad (5.27)$$

Овде је ω ковекторско поље (нпр. брзина третирана као 1-форма), а транспонована Јакобијева матрица $(\mathbf{J}_\Psi)^T$ обезбеђује исту линеарну комбинацију компоненти као и горња двострука сума. Свака компонента $(\Psi^* \omega)_i$ добија се множењем редова транспоноване Јакобијеве матрице вектором $\omega(\Psi(\mathbf{x}))$. Ова еквивалентност потиче из саме дефиниције матричног множења. Дакле, док оператор гурања делује преко Јакобијеве матрице $\mathbf{J}_\Psi$ на векторе, повлачење делује преко транспоноване матрице $(\mathbf{J}_\Psi)^T$ на ковекторе. Оба оператора су фундаментална за геометријски тачну симулацију течности у оквиру семи-Лагранжове методе. Повлачење је „спољашња” верзија промене променљивих у интегралном рачуну. У контексту флуида, оно обезбеђује повезивање Лагранжових и Ојлерових координата.

Ψ се дефинише експлицитно као инверзно пресликавање флуидног тока $\Psi_t(\mathbf{x}) : M_t \rightarrow M_0$ [11]. У складу са (5.20), Ψ_t се добија као решење обичне диференцијалне једначине:

$$\frac{\partial \Psi_t(\mathbf{x})}{\partial t} = -\mathbf{J}_{\Psi_t}(\mathbf{x}) \mathbf{u}(\mathbf{x}), \quad \Psi_0(\mathbf{x}) = \mathbf{x}. \quad (5.28)$$

У дискретизованој форми, ова једначина се може решавати експлицитним Ојлеровим поступком, што најчешће није довољно прецизно, па се због тога обично користе методе Рунге-Кута трећег и четвртог реда [33].

Као што је раније приказано, семи-Лагранжова метода за адвекцију вектора брзине $\mathbf{u}$ може се представити у облику:

$$\mathbf{u}^{\text{adv}}(\mathbf{x}) = \mathbf{u}(\Psi_t(\mathbf{x})), \quad (5.29)$$

док ковекторски приступ, на основу (5.27), даје:

$$\eta^{\text{adv}}(\mathbf{x}) = (\Psi_t^* \eta)(\mathbf{x}) = (\mathbf{J}_{\Psi_t}(\mathbf{x}))^T \eta(\Psi_t(\mathbf{x})), \quad (5.30)$$

за свако $\mathbf{x} \in N$, где је $\eta = \mathbf{u}^\flat$ ковектор који одговара вектору брзине. Овај израз може послужити као полазна релација за нумеричку имплементацију ковекторске адвекције.

5.3 Лијев извод и Лијева адвекција

Лијев извод векторских и ковекторских поља

Лијев извод (енг. *Lie derivative*) је фундаментални оператор у диференцијалној геометрији који мери брзину промене различитих поља дуж неког другог векторског поља. Нека је $\mathbf{v}$ векторско поље на многострукости M , и нека је Φ_t једнопараметарски ток, односно дифеоморфизам који помера геометријски објекат T (скаларно, векторско или поље диференцијалних форми) дуж интегралних крива поља $\mathbf{v}$, у зависности од параметра t . Лијев извод објекта T се дефинише као:

$$\mathcal{L}_{\mathbf{v}}T = \frac{d}{dt} [\Phi_t^*T]_{t=0}, \quad (5.31)$$

где је Φ_t^*T повлачење објекта T дуж Φ_t .

У случају скаларних поља, односно диференцијалних 0-форми, Лијев извод постаје усмерен извод, односно извод у правцу $\mathbf{v}$. Када су у питању векторска поља ситуација је мало сложенија јер се за њих оператор повлачења не може дефинисати. Нека је M диференцијабилна многострукост и нека су $\mathbf{v}$ и $\mathbf{w}$ два глатка векторска поља на M . Тада се Лијев извод векторског поља $\mathbf{w}$ у смеру $\mathbf{v}$ у тачки $p \in M$, означен са $\mathcal{L}_{\mathbf{v}}\mathbf{w}$, дефинише као:

$$(\mathcal{L}_{\mathbf{v}}\mathbf{w})_p = \lim_{t \rightarrow 0} \frac{(\Phi_{-t})_* \mathbf{w}_{\Phi_t(p)} - \mathbf{w}_p}{t} = \frac{d}{dt} [(\Phi_{-t})_* \mathbf{w}_p]_{t=0}, \quad (5.32)$$

где је $(\Phi_{-t})_*$ *пурање* дифеоморфизма Φ_t , које пресликава тангентни вектор у тачки $\Phi_t(p)$, у тангентни вектор у p . Ова дефиниција каже да се вектор $\mathbf{w}$ у близини тачке p „вуче назад” преко тока Φ_t , тако да се може упоредити са вредношћу $\mathbf{w}_p$ у самој тачки p . Разлика у граничној вредности даје брзину промене поља $\mathbf{w}$ дуж тока поља $\mathbf{v}$.

За диференцијалну k -форму ω , Лијев извод у смеру векторског поља $\mathbf{v}$ добија се на основу (5.31), коришћењем повлачења:

$$(\mathcal{L}_{\mathbf{v}}\omega)_p = \frac{d}{dt} [\Phi_t^* \omega_p]_{t=0} = \lim_{t \rightarrow 0} \frac{\Phi_t^* \omega_{\Phi_t(p)} - \omega_p}{t}. \quad (5.33)$$

Овде се форма ω_p у тачки $\Phi_t(p)$ повлачи на тачку p оператором повлачења Φ_t^* , тако да се може упоредити са ω_p .

Лијев извод је инваријантан при координатним трансформацијама и задовољава Лајбницово правило за производе:

$$\mathcal{L}_{\mathbf{v}}(\alpha \wedge \beta) = \mathcal{L}_{\mathbf{v}}\alpha \wedge \beta + \alpha \wedge \mathcal{L}_{\mathbf{v}}\beta. \quad (5.34)$$

Он је, такође, компатибилан са протоком: ако је $\mathcal{L}_v \omega = 0$, онда је форма ω инваријантна дуж тока поља $\mathbf{v}$. Иако је Лијев извод по природи геометријски и координатно независан појам, у тродимензионалном еуклидском простору $\mathbb{R}^3$ са стандардном метриком, његово деловање на ковекторско поље може бити прецизно и интуитивно описано коришћењем операција векторске анализе: градијента, ротора и дивергенције. Нека је $\mathbf{u}^b$ 1-форма која одговара векторском пољу $\mathbf{u}$, и нека је $\mathbf{v}$ друго векторско поље, које дефинише смер у ком рачунамо Лијев извод. У $\mathbb{R}^3$, овај извод може бити преведен у векторски облик. Ако са $\mathbf{w}$ означимо векторско поље које одговара Лијевом изводу $\mathcal{L}_v \mathbf{u}^b$ (тј. $\mathbf{w} = (\mathcal{L}_v \mathbf{u}^b)^\sharp$), тада важи следећа једнакост у векторској анализи:

$$\mathbf{w} = (\mathcal{L}_v \mathbf{u}^b)^\sharp = (\mathbf{v} \cdot \nabla) \mathbf{u} + (\nabla \mathbf{v})^T \cdot \mathbf{u}, \quad (5.35)$$

у коме први члан, $(\mathbf{v} \cdot \nabla) \mathbf{u}$, представља извод вектора $\mathbf{u}$ у смеру вектора $\mathbf{v}$ и описује како се векторско поље $\mathbf{u}$ мења дуж смера кретања дефинисаног са $\mathbf{v}$. Други члан, $(\nabla \mathbf{v})^T \cdot \mathbf{u}$, узима у обзир *деформацију шока*, тј. како се суседне линије тока разилазе, ротирају или растежу. Овде је $\nabla \mathbf{v}$ градијент векторског поља (Јакобијева матрица), а транспонована верзија се појављује због природе повлачења ковектора.

Кључно је напоменути да Лијев извод ковектора није исто што и материјални извод вектора. У контексту флуида, ако је $\mathbf{u}$ брзина честица флуида, класични материјални извод $D\mathbf{u}/Dt$ садржи само члан $(\mathbf{v} \cdot \nabla) \mathbf{u}$, док је Лијев извод $\mathcal{L}_v \mathbf{u}$ дат са (5.35), и разликује се од њега ако је $\mathbf{v} \neq \mathbf{u}$. Међутим, када је $\mathbf{v} = \mathbf{u}$ тада је $\mathcal{L}_u \mathbf{u} = 0$. Онда Лијев извод $\mathcal{L}_u \mathbf{u}^b$ описује како се ковекторско поље $\mathbf{u}^b$ мења дуж самог себе, што је централно за анализу адвекције у симулацијама течности.

Лијева адвекција

Како смо већ напоменули, у традиционалним симулацијама нестишљивих флуида, поље брзине $\mathbf{v}$ се адвектује коришћењем материјалног извода $D\mathbf{u}/Dt$, што одговара померању вектора дуж тока, али не гарантује очување вртложних структура. Основна идеја је да се брзина третира као диференцијална 1-форма $\mathbf{u}^b$ (тј. ковекторско поље) [11]. Разлика се појављује у начину како се ова величина мења са временом. Уместо да се користи материјални извод, о чему смо говорили у претходним поглављима, у раду је коришћена тзв.

Лијева адвекција, према којој се $\mathbf{u}^b$ мења у времену у складу са оператором повлачења дуж тока течности.

Нека је Φ_t дифеоморфизам који описује еволуцију честица течности за време t уз ток векторског поља $\mathbf{u}$. Тада је промена вредности 1-форме $\mathbf{u}^b$ у временском интервалу Δt , на основу (5.30), дата са:

$$\mathbf{u}^b(t + \Delta t) = \Psi_{\Delta t}^* \mathbf{u}^b(t) = \Phi_{-\Delta t}^* \mathbf{u}^b(t). \quad (5.36)$$

На основу (5.33) и (5.36), Лијев извод од $\mathbf{u}^b$ можемо написати у облику:

$$\mathcal{L}_{\mathbf{v}} \mathbf{u}^b = \frac{d}{dt} [\Phi_t^* \mathbf{u}^b]_{t=0} = \lim_{\Delta t \rightarrow 0} \frac{\Phi_{\Delta t}^* \mathbf{u}^b - \mathbf{u}^b}{\Delta t}. \quad (5.37)$$

Онда је временска еволуција поља $\mathbf{u}^b$ уз ток $\mathbf{v}$ управо дата Лијевим изводом:

$$\frac{\partial \mathbf{u}^b}{\partial t} = -\mathcal{L}_{\mathbf{v}} \mathbf{u}^b. \quad (5.38)$$

Предност овог приступа је да Лијева адвекција 1-форме аутоматски задовољава Келвинову теорему циркулације. Циркулација 1-форме $\mathbf{u}^b$ дуж затворене криве C дефинише се као:

$$\Gamma = \oint_C \mathbf{u}^b. \quad (5.39)$$

Ако се крива C креће заједно са током течности (тј. еволуира у $\Phi_t(C)$), тада из (5.25) и (5.30) директно следи:

$$\oint_{\Phi_t(C)} \mathbf{u}^b(t) = \oint_C \Phi_t^* \mathbf{u}^b(t) = \oint_C \mathbf{u}^b(0). \quad (5.40)$$

Ово значи да се циркулација очувава током времена, што тврди Келвинова теорема за идеалну (невискозну) течност. Насупрот томе, традиционална адвекција вектора не гарантује очување циркулације. Проблем настаје зато што традиционални приступ не узима у обзир деформацију тока, дефинисану Јакобијевом матрицом, а која се појављује у Лијевом изводу (5.35). Као последица, долази до губитка циркулације, па корекција (пројекција) притиска мора да коригује додатне поремећаје дивергенције због чега се вртложност смањује и вртлози постепено нестају.

Ојлерове једначине у ковекторској форми

Ојлерове једначине за невискозну, нестишљиву течност се могу преформулисати у ковекторску форму, за коју смо видели да омогућава природно

очување циркулације и вртложних структура. Полазна тачка је класична Ојлерова једначина за поље брзина $\mathbf{u}$:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p, \quad (5.41)$$

у којој смо на десној страни занемарили деловање спољашњих сила. Додавањем члана $(\nabla \mathbf{u}) \cdot \mathbf{u} = (1/2) \nabla |\mathbf{u}|^2$ на обе стране ове једначине добија се:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + (\nabla \mathbf{u}) \cdot \mathbf{u} = -\frac{1}{\rho} \nabla \left(p - \frac{\rho}{2} |\mathbf{u}|^2 \right). \quad (5.42)$$

Први корак је да се брзина $\mathbf{u}$ не третира као векторско, већ као ковекторско поље $\mathbf{u}^b$. У овом формализму се Ојлерова једначина може написати у облику:

$$\frac{\partial \mathbf{u}^b}{\partial t} + \mathcal{L}_{\mathbf{u}} \mathbf{u}^b = -\frac{1}{\rho} d\lambda, \quad (5.43)$$

где је $\mathcal{L}_{\mathbf{u}}$ Лијев извод дуж векторског поља $\mathbf{u}$, а десна страна је диференцијал скаларног поља који се назива и Лагранжовим притиском $\lambda = p - \rho |\mathbf{u}|^2/2$. Лева страна ове једначине представља промену линијског интеграла $\int_{\gamma} \mathbf{u}^b$ дуж криве γ која се и сама креће са током флуида, што је директна генерализација Келвинове теореме.

Ово нам омогућава да унесемо одговарајуће промене у алгоритам за нумеричко решавање Ојлерових једначина описаног у одељку 3.3. Први корак алгоритма, којим се израчунавају вредности брзине $\mathbf{u}^{n+1}$ у тренутку t_{n+1} , је адвекција, којом се на основу брзина из претходног временског тренутка t_n одређује привремена вредност $\mathbf{u}^A$. Заправо, једино је потребно изменити овај корак. Уместо да користимо материјални извод, ковекторско поље ћемо адвектовати помоћу једначине (5.38):

$$\frac{\partial \mathbf{u}^b}{\partial t} + \mathcal{L}_{\mathbf{u}^n} \mathbf{u}^b = 0, \quad (5.44)$$

чије је решење дефинисано повлачењем (5.30) помоћу инверзног тока $\Psi_t = \Phi_{-t}$:

$$\mathbf{u}^b(\mathbf{x})^A = (\Psi_t^* \mathbf{u}^b)(\mathbf{x}) = (J_{\Psi_t}(\mathbf{x}))^T \mathbf{u}^b(\Psi_t(\mathbf{x}), t_n). \quad (5.45)$$

Након тога примењује се стандардна пројекција (корекција) притиска, којом се обезбеђује услов нестишљивости. Ова промена се може једноставно имплементирати у постојеће кодове засноване на Ојлеровој методи заменом семи-Лагранжове адвекције са повлачењем ковектора. Практично то значи

да се ковекторска адвекција, на основу релације (5.45) реализује у два корака. Први је да се, на пример, семи-Лагранжовим поступком израчуна $\mathbf{u}^b(\Psi_t(\mathbf{x}), t_n)$. Затим се коришћењем брзина из претходног временског корака t_n израчунава транспоновани Јакобијан трансформације $(J_{\Psi_t}(\mathbf{x}))^\top$ и њиме коригује брзина из претходног корака. Ова корекција се примењује локално, што значајно поједностављује њену имплементацију.

Веома важан квалитет ковекторског приступа лежи у чињеници да адвекција ковектора имплицитно решава и једначину кретања флуида у вртложној формулацији (2.17) (наравно за случај када је флуид невискозан, тј. када важи $\mu = 0$). Показано је да ако $\mathbf{u}^b$ задовољава једначину (5.44), онда вртложност $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ задовољава једначину (2.17) [11]. Кључни корак у доказу је чињеница да Лијев извод комутира са спољашњим изводом када је $\nabla \cdot \mathbf{v} = 0$: $\mathcal{L}_{\mathbf{v}}(d\mathbf{u}^b) = d(\mathcal{L}_{\mathbf{v}}\mathbf{u}^b)$. Дакле, ако је $d(\mathcal{L}_{\mathbf{v}}\mathbf{u}^b) = -\partial_t(d\mathbf{u}^b)$, онда је: $\mathcal{L}_{\mathbf{v}}(d\mathbf{u}^b) = -\partial_t(d\mathbf{u}^b)$, што уз $\boldsymbol{\omega}^b = d\mathbf{u}^b$ доводи до:

$$\frac{\partial \boldsymbol{\omega}^b}{\partial t} + \mathcal{L}_{\mathbf{v}}\boldsymbol{\omega}^b = 0, \quad (5.46)$$

односно до једначине која је еквивалентна једначини (2.17), где је $\boldsymbol{\omega}^b$ диференцијална 2-форма вртложности. Другим речима, ковекторски метод имплицитно решава вртложну једначину на нивоу брзине, без експлицитне промене променљивих на $\boldsymbol{\omega}$. При томе није потребно спроводити нумерички веома захтеван поступак реконструкције поља вектора брзине на основу израчунатих вртложности.

Значајна предност ковекторског приступа је и то што адвекција ковектора комутира са пројекцијом притиска, чиме се елиминишу грешке због раздвајања оператора које се јављају у традиционалним методима (видети поглавље 3). Ово је у директној супротности са векторским приступом, у коме пројекција може да промени циркулацију јер не комутира са операцијом адвекције. На тај начин, ковекторска метода осигурава дуже трајање и стабилније вртлоге, чак и без експлицитног моделовања вртложних структура.

Веза ковекторске и импулсне методе

Основни геометријски разлог за разликовање брзине и импулса лежи у њиховом понашању при промени координата. Брзина припада тангентном простору $T_p M$ многострукости конфигурација, што значи да се њене компо-

ненте трансформишу контраваријантно. Зато је она контраваријантни вектор – у физичком смислу, она показује правац и интензитет кретања дуж путање. С друге стране, генералисани импулс (конјугован координати) припада котангентном простору T_p^*M и представља ковектор (1-форму). Његове компоненте се трансформишу коваријантно, што је неопходно да би скаларни производ $p_i v^i$ (који има физички смисао, нпр. енергија) био инваријантан у односу на избор координата. Овај дуалитет је кључан за Хамилтонов формализам и изградњу фазног простора као котангентног раслојења, где једноставна симплектичка структура $dq^i \wedge dp^i$ постоји управо захваљујући супротним законима трансформације координата $\mathbf{q}$ и импулса $\mathbf{p}$.

И ковекторска метода [11] и импулсна метода [42] полазе од истог геометријског увида – да брзина $\mathbf{u}$ није права величина коју треба адвектовати ако желимо да сачувамо вртлоге. Уместо тога, оне уводе помоћно поље (импулс $\mathbf{m} = \mathbf{u} + \nabla \varphi$ у првој, ковектор $\mathbf{u}^b$ у другој), које се адвектује једначином облика $\partial(\cdot)/\partial t + (\mathbf{u} \cdot \nabla)(\cdot) = ((\cdot) \cdot \nabla)\mathbf{u}$. Ова једначина гарантује да се циркулација (а самим тим и вртложност) чува на начин који поштује Келвинову теорему. У оба приступа, након адвекције следи реконструкција физичке брзине $\mathbf{u}$ решавањем Поасонове једначине – што је рачунски најзахтевнији део. Обе методе су развијене првенствено за симулацију идеалног (невискозног) флуида и показују далеко боље очување вртлога од стандардних „адвекција-пројекција“ решавача.

Импулсна метода третира $\mathbf{m}$ као обично векторско поље (са компонентама у координатној бази) и након адвекције увек захтева глобално решавање Поасонове једначине $\nabla^2 \varphi = \nabla \cdot \mathbf{m}$ на читавом домену да би се добило φ , а затим $\mathbf{u} = \mathbf{m} - \nabla \varphi$. То је рачунарски веома захтевно, посебно на финим мрежама. Ковекторска метода, међутим, експлицитно користи чињеницу да је $\mathbf{u}^b$ ковектор (1-форма). Ова геометријска природа омогућава да се адвекција изведе тако да $\mathbf{u}^b$ остане „близу“ недивергентног поља, па је потребна само локална корекција уместо глобалног Поасоновог решавача. У пракси, то значи да се ковекторска метода може имплементирати са свега неколико линија кода додатих на постојећи „адвекција-пројекција“ решавач, док импулсна метода захтева потпуно нови, сложенији нумерички оквир. Стога је ковекторска метода бржа и једноставнија за имплементацију, уз исте или боље перформансе у очувању вртлога. Зато се данас сматра практичнијим наследником импулсне методе.

Глава 6

Имплементација флуидног симулатора

Један од циљева овог мастер рада је развој тродимензионалног кода за нумеричку симулацију струјања нестипљивих и невискозних флуида са слободним површинама, заснованог на теоријским основама представљеним у претходним поглављима. Код је реализован као модуларни, објектно-оријентисани систем у програмском језику C++, који се стандардно користи за рачунарске захтевне симулације. Основни циљ примењене архитектуре је одвајање логике дискретизације простора (прорачунске мреже) од физичких алгоритама (решавача за адвекцију и пројекцију притиска), што омогућава лакше тестирање, одржавање и једноставније проширење кода. У том циљу, примењени су наслеђивање и полиморфизам за руковање различитим типовима мрежа.

Имплементација доследно прати концепте и препоруке за симулацију флуида описане у књигама [2] и [28]. Код се темељи искључиво на Ојлеровом приступу са семи-Лагранжовом адвекцијом брзина на МАС мрежи, као и на коекторској корекцији. Иако није директно заснован на јавно доступној библиотеци за развој флуидних кодова за анимацију и визуелне ефекте [19], имплементирани код задржава сличну номенклатуру класа и приступ прорачунима као у референци [28].

Код (доступан на <https://github.com/domeGIT/CovectorFluidSolver>) има укупно око 6000 линија и организован је у три логичке целине - модула:

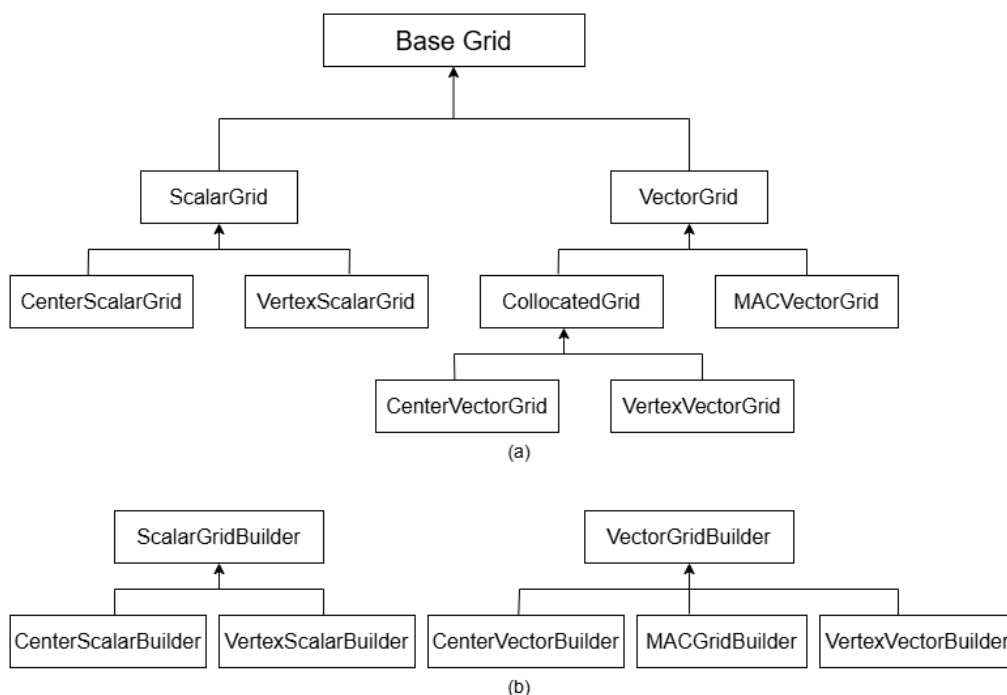
1. Модул за генерисање и управљање прорачунским мрежама чије класе дефинишу структуре података за чување физичких величина на њима.
2. Модул за физичке решаваче (солвере) чије класе имплементирају алго-

ритме за адвекцију поља брзина и *функција ниво скуја* (енг. *level set function*) за опис слободне површине, као и пројекцију притиска ради очувања недивергентности.

3. Помоћни модул обухвата функције за интерполацију и екстраполацију, као и *градитељске* (енг. *builder*) класе за креирање мрежа и иницијализацију поља брзине и скаларних функција, чиме се логика иницијализације одваја од логике симулације ради боље читљивости кода.

6.1 Прорачунске мреже

За дискретизацију простора коришћене су фиксне декартовске мреже. Имплементација раздваја смештај скаларних и векторских величина како би се избегле нумеричке нестабилности. Слика 6.1 приказује а) дијаграм мрежних класа (које су детаљније описане у одељку 3.2) коришћених у прорачунима и б) одговарајуће градитељске класе.



Слика 6.1: а) Мрежне класе и б) њихови градитељи.

`BaseGrid` је апстрактна базна класа која дефинише заједнички интерфејс за све типове мрежа. `ScalarGrid` и `VectorGrid` су специјализоване класе за

скаларна (притисак, густина) и векторска поља (брзина). `MACVectorGrid` и `CenterScalarGrid` су класе које имплементирају специфичне распореде података на мрежи. За смештање поља брзине задужена је класа `MACVectorGrid`, што значи да су компоненте брзине (u, v, w) дефинисане на страницама ћелија, док су скаларне величине (притисак, густина) дефинисане у њиховим центрима (`CenterScalarGrid`). Овакав распоред омогућава природну дискретизацију дивергенције и градијента без потребе за додатном интерполацијом у основним корацима, што смањује грешке заокруживања.

Приступ елементима мреже је оптимизован како би се минимизирали промашаји *кеш меморије* (енг. *cache miss*). Користи се линеаризација 3D индекса у 1D низ (нпр. `index = x + y * Nx + z * Nx * Ny`) унутар класа. Класа `MACVectorGrid` имплементира методе `sample()` и `cubicSample()` које користе трилинеарну, односно кубну интерполацију (преко `Utils` модула) за добијање вредности на произвољним позицијама у домену. Методе `fillData()` прихватају `std::function` као аргумент, што омогућава флексибилно попуњавање поља произвољним аналитичким изразима или процедуралним генераторима. Овом модулу припада и класа `GridManager` која је одговорна за иницијализацију, алокацију меморије и животни циклус објеката мреже.

Свим ћелијама мреже придружени су маркери (ознаке) којима је одређено шта се у њима налази. Они су дефинисани посебном `enum` структуром:

```
enum Medium
{
    FLUID = 0,
    AIR = 1,
    COLLIDER = 2
};
```

Садржај ћелија са маркерима `FLUID` и `AIR` у иницијалном стању се може мењати у току времена, па се сходно томе мењају и маркери. Маркери ћелија на почетку означених са `COLLIDER` су фиксни, с обзиром на то да такве ћелије описују чврсте препреке кретању флуида. Ћелије које садрже флуид и препреку описане су функцијама ниво скупа `fluidSDF` и `colliderSDF`, респективно.

6.2 Решавачи

Имплементирани решавачи чине алгоритмско језгро за симулацију кретања флуида на основу Навије–Стоксових једначина на фиксним декартовским мрежама. Кључни решавачи укључују: `SemiLagrangianAdvectSolver` за адвекцију векторских и скаларних поља, као и ковекторску корекцију ради очувања вртложности, те `PressureSolver` за пројекцију брзине у недивергентно поље решавањем Поасонове једначине. Решавачи су дизајнирани са јасним интерфејсима и подршком за OpenMP паралелизацију, што осигурава скалабилност и флексибилност кода.

Класа `GridFluidSolver` има улогу оркестратора који координира модуле за управљање мрежама, адвекцију слободне површине и брзина, као и пројекцију притиска. Ова класа наслеђује базни интерфејс из класе `FluidSolver` и специјализована је за рад са `MACVectorGrid` и `CenterScalarGrid` објектима, уз управљање временском петљом симулације. Конструктор `GridFluidSolver` врши следеће кључне операције:

1. *Иницијализацију мрежа*: Креирање класе `GridManager` са задатом резолуцијом, координатним почетком и кораком мреже.
2. *Регистрацију поља*:
 - „`FREE_SURFACE`”: `CenterScalarGrid` за функцију ниво скупа (слободна површина);
 - „`VORTICITY`”: `CenterScalarGrid` за праћење вртложности (за визуелизацију или будуће проширење).
3. *Дефинисање препреке*:
 - Креира се класа `colliderSDF` која представља препреку као инстанцу класе `CenterScalarGrid`, иницијализовану на вредност `std::numeric_limits<double>::max()`;
 - За `COLLIDER = true` дефинише се функција ниво скупа `colliderSDF`.

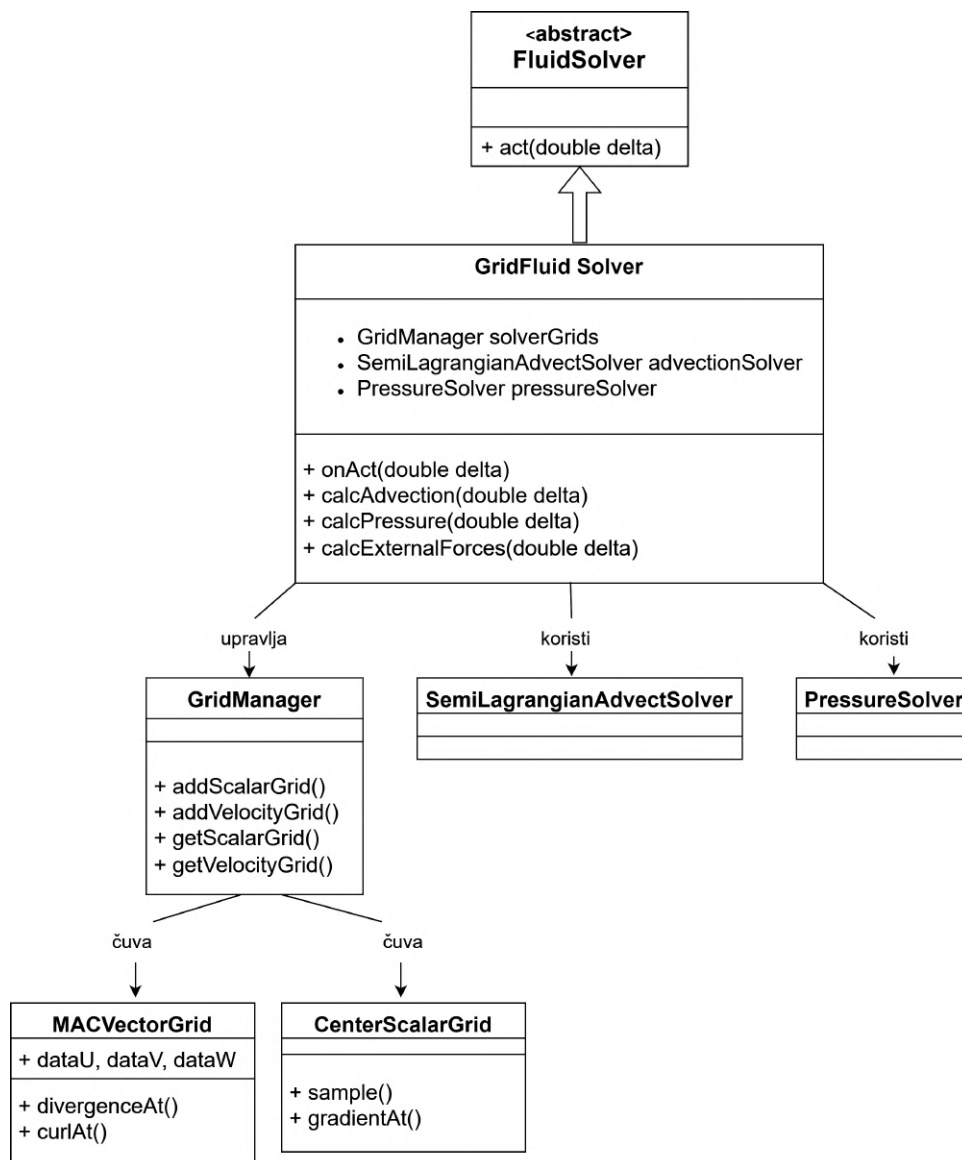
Главни алгоритам је имплементиран у методи `onAct(double time_step)`, која се извршава за сваки временски корак (`time_step`). Овде ћемо приказати код методе `onAct()`, с обзиром на њену једноставност:

```
void GridFluidSolver::onAct(double time_step)
{
    preAct(time_step);
    calcExternalForces(time_step);
}
```

```

    calcPressure(time_step);
    calcAdvection(time_step);
    postAct(time_step);
} .

```



Слика 6.2: Дијграм класа решавача за симулацију флуида.

Метода `preAct(time_step)` је први корак и она за сада врши само валидацију мреже. Следећи корак је урачунавање деловања силе гравитације на флуид, које реализује метода `calcExternalForces(time_step)`. Трећи корак је решавање Пуасонове једначине за притисак и одговарајућа корекција бр-

зина, што је имплементирано у методи `calcPressure(time_step)`. Затим следи адвекциони корак, реализован у методи `calcAdvection(time_step)` која је најкомплекснија. Прва фаза је екстраполација флуидног поља брзина у ваздуху, а затим се на основу CFL услова процењује да ли је потребно временски корак поделити на више поткорака. Након тога, семи-Лагранжовом методом се врши адвекција функције ниво скупа и флуидног поља брзина, у претходно одређеном броју поткорака. У овој фази се опционо врши и ковекторска корекција флуидних брзина. На крају другог, трећег и четвртог корака, у којима долази до промена поља брзина, позива се метода `applyBoundaryCondition()` која осигурава важење граничних услова. У последњем кораку `postAct(time_step)` врши се реиницијализација функције ниво скупа.

Адвекција

`SemiLagrangianAdvectionSolver` је најсложенија класа у систему, декларисана у заглављу `SemiLagrangianAdvectionSolver.h` и садржи стандардну семи-Лагранжову адвекцију за векторска и скаларна поља, као и ковекторску корекцију. Метода `GridFluidSolver::calcAdvection` координира стандардну и ковекторску адвекцију. Ковекторска корекција се искључује у следећим случајевима (када се алгоритам враћа на стандардну семи-Лагранжову методу):

- Чвор је у ваздушном региону ($\text{fluidSDF} > h$). Вредност h је дефинисана у `SemiLagrangianAdvectionSolver::advection` методи;
- Чвор је близу слободне површине ($-2h < \text{fluidSDF} < 2h$) — спречава нестабилност на граници;
- Јакобијан $\mathbf{J}$ је сингуларан ($|\det(\mathbf{J})| < 1e^{-8}$) — спречава нумеричке дивергенције.

Адвекција слободне површине пре адвекције брзине осигурава конзистентност. За очување монотоности решења, након ковекторске корекције примењује се тзв. BFESS ограничавач (лимитер) који пажљиво обрађује чворове близу слободне површине, проверавајући да ли су суседне ћелије у флуиду пре него што их укључи у прорачунски *мрежни шаблон* (енг. *stencil*). Иако имплементиран у коду, ограничавач није побољшао квалитет решења.

У оквиру адвекције имплементирани у класи `SemiLagrangianAdvectionSolver` кључни корак за очување вртложних структура представља ковекторска корекција. Она се заснива на трансформацији поља брзине дуж карактеристика

тока према изразу (5.45), $\mathbf{u}^{n+1}(\mathbf{x}) = (\mathbf{D}\Psi)^\top \cdot \mathbf{u}^n(\Psi(\mathbf{x}))$, где је Ψ инверзно пресликавање које пресликава позицију $\mathbf{x}$ у времену t^{n+1} уназад дуж линије тока до позиције $\Psi(\mathbf{x})$ у времену t^n , а $\mathbf{D}\Psi$ означава Јакобијан овог пресликавања. У практичној имплементацији, Јакобијан се апроксимира методом централних коначних разлика: за сваку тачку $\mathbf{x} = (x, y, z)$ на мрежи, врши се *нумеричко итражење уназад* (енг. *backtracing*) за шест пертурбованих позиција $\mathbf{x} \pm h\mathbf{e}_i$ (где је $h = 0.1 \cdot \Delta x$ мали корак пертурбације, а $\mathbf{e}_i$ јединични вектори координатних оса), чиме се добијају пресликане позиције $\Psi(\mathbf{x} \pm h\mathbf{e}_i)$. Колоне матрице $\mathbf{D}\Psi$ се затим израчунавају као $\frac{\partial \Psi}{\partial x_i} \approx \frac{\Psi(\mathbf{x} + h\mathbf{e}_i) - \Psi(\mathbf{x} - h\mathbf{e}_i)}{2h}$, након чега се формира транспонована матрица заменом индекса $[\mathbf{D}\Psi]_{ij}^\top = [\mathbf{D}\Psi]_{ji}$.

Имплементација функције `compute_jacobian_transpose` укључује неколико нивоа заштите од нумеричке нестабилности: (1) провера да ли се тачка налази у флуиду преко функције ниво скупа ($\phi < 0$); (2) искључивање корекције близу слободне површине ($|\phi| < 2h$) где би пресликавање могло бити сингуларно; (3) верификација да све пертурбоване тачке остају у домену флуида; (4) провера детерминанте Јакобијана ($|\det(\mathbf{D}\Psi)| > 10^{-4}$), како би се избегло дељење са скоро сингуларним матрицама. У случају да било која од ових провера не успе, алгоритам се враћа на стандардну семи-Лагранжову адвекцију коришћењем јединичне трансформације $(\mathbf{D}\Psi)^\top \approx \mathbf{I}$, чиме се осигурава робустност симулације без компромитовања стабилности. Овај приступ, иако рачунски захтевнији од класичне адвекције (захтева 7 операција нумеричко тражење уназад по чвору уместо једне), омогућава значајно боље очување кохерентних вртложних структура, што је кључно за симулације попут торусног вртлога или интеракције млазева.

Контрола временског корака

У методи `GridFluidSolver::calcAdvection`, имплементиран је механизам за адаптивно потстепеновање ради задовољења CFL услова. Предности овог приступа су:

- *Стабилност*: спречава нумеричку нестабилност при променама брзине;
- *Ефикасност*: користи минималан број поткорака када је то могуће;
- *Робустност*: горња граница од 8 поткорака спречава неконтролисани раст времена израчунавања.

Реиницијализација функције ниво скупа

Да би се одржао услов $|\nabla\phi| = 1$ (видети одељак 4.3), функција ниво скупа `fluidSDF` се периодично реиницијализује. Овај процес спроводи функција `GridFluidSolver::reinitializeFreeSurface` чије су карактеристике:

- стабилан псеудо-временски корак $\Delta\tau$ се израчунава на основу минималног корака мреже,
- користе се схеме узводних разлика за израчунавање парцијалних извода $\partial\phi/\partial x$, $\partial\phi/\partial y$, $\partial\phi/\partial z$,
- еволуциона једначина (4.12) се решава коришћењем `min/max` операција на *upwind* изводима,
- користи се двоструки бафер (`tempGrid` и `outputMesh`) за избегавање зависности од редоследа ажурирања.

Екстраполација брзина у ваздуху

Метод `GridFluidSolver::extrapolateVelocityToAir` обезбеђује да брзине у ваздушним ћелијама буду конзистентне са брзинама на граничној површини, што је кључно за стабилност адвекције скалара. Алгоритам садржи следеће кораке:

1. *Класификација*: за сваку компоненту брзине одређује се да ли је тачка у флуиду (`fluidSDF < 0`);
2. *Иницијализација*: брзине ван флуида се постављају на нулу;
3. *Екстраполација*: позива се функција `Utils::extrapolateToRegion` са вредношћу параметра `sf = 5.0`, чиме се омогућава брза екстраполација константне вредности дуж нормале на граничну површину;
4. *Ажурирање*: екстраполиране вредности се уписују назад у `MACVectorGrid`.

Гранични услови

Систем секвенцијално примењује два нивоа граничних услова:

1. *Домен* (`GridFluidSolver::applyDomainBoundaryConditions`):
 - Услови слободног клизања на свим зидовима домена;
 - Нормална компонента брзине се поставља на нулу на ивицама мреже;
 - Примењује се независно за сваку компоненту (u, v, w) на одговарајућим страницама.
2. *Прејрека* (`GridFluidSolver::applyColliderBoundaryConditions`):

- Користи функцију ниво скупа `colliderSDF` за представљање геометрије препрека;
- За сваку тачку мреже унутар препреке (`colliderSDF < 0`):
 - израчунава се нормала на површини као градијент функције нивоа `fluidSDF`: $\mathbf{n} = \nabla \phi / |\nabla \phi|$,
 - пројектује се вектор брзина на нормалу на површини: $v_n = \mathbf{u} \cdot \mathbf{n}$,
 - ако се флуид креће ка површини ($v_n < 0$), нормална компонента се одузима: $\mathbf{u}_{new} = \mathbf{u} - v_n \mathbf{n}$.

Пројекција притиска

Решавањем Поасонове једначине (3.24), записане у облику система линеарних једначина (3.29), обезбеђује се да поље брзина добијено у претходном кораку буде недивергентно. Ово реализује функција `PressureSolver::solve` у четири фазе:

1. Функција `PressureSolver::setupMarkers` поставља маркере и граничне услове. Систем користи тростепени систем маркера (`COLLIDER`, `FLUID`, `AIR`) дефинисан на основу функција ниво скупа `fluidSDF` и `colliderSDF`. За ћелије означене као `COLLIDER` примењују се Нојманови гранични услови, што се имплементира постављањем дијагоналног елемента на јединицу и изједначавањем десне стране једначине са нулом;
2. Функција `PressureSolver::setupSystem` користи `Eigen::Triplet` за формирање матрице система (`Eigen::SparseMatrix<double>`) и вектора десне стране (`Eigen::VectorXd`);
3. Добијени систем се решава применом `Eigen::ConjugateGradient` решавача, који користи симетрију матрице (`Eigen::Upper`). Овај метод је ефикасан за велике, ретке и симетрично позитивно дефинисане системе типичне за симулације флуида;
4. Након добијања вредности притиска, брзине се коригују уз помоћ функције `PressureSolver::applyPressureGradient`.

Паралелизација и управљање меморијом

Независна обрада чворова мреже при иницијализацији и примени спољашњих сила (енг. *embarrassingly parallel*) омогућава ефикасну паралелизацију посредством OpenMP директива. Критичне петље у модулима `PressureSolver`,

`SemiLagrangianAdvectionSolver` и `MACVectorGrid` (ковекторска корекција, адвекција методом `forEachU/V/W()` и конверзија података за интерполацију) паралелизоване су директивом `#pragma omp parallel for`. Овим се постиже скоро линеарно убрзање и смањење времена израчунавања на вишејезгарним системима, посебно код мрежа високе резолуције.

Централизовано управљање меморијом посредством класе `GridManager` минимизира ризик од цурења меморије и олакшава реалокацију при промени резолуције. Стратегија предлокације (преко `MACVectorGrid::onResize()` и `GridManager`) осигурава једнократну алокацију на почетку, елиминишући динамичку алокацију у главној временској петљи. Велики објекти се прослеђују путем паметних показивача (`std::shared_ptr`) како би се избегло копирање, које се врши само по потреби (нпр. `velInput` за притисак) користећи конструкторе копирања.

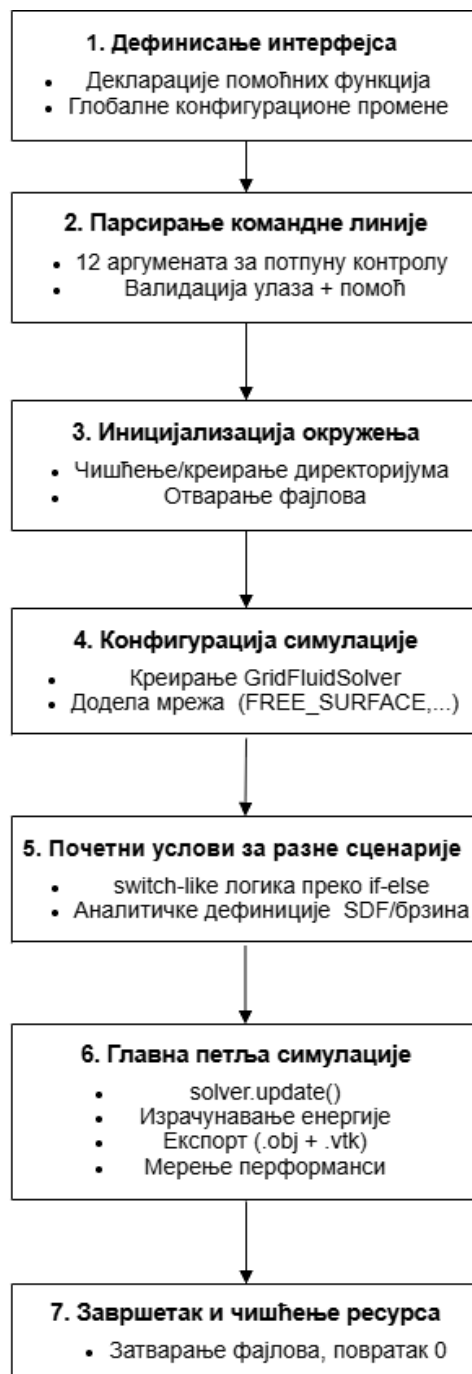
Имплементација садржи више нивоа заштите од нумеричких проблема:

- `safe_val()` функције: спречавају ширење NaN/Inf вредности;
- `std::clamp`: сви индекси и интерполациони параметри се експлицитно ограничавају пре приступа меморији;
- `assert` провере: валидација конзистентности параметара мреже у `preAct` методи.

Главна функција (`main.cpp`)

Функција `main` представља улазну тачку апликације и координира ток симулације: од иницијализације домена и почетних услова, преко временске еволуције, до визуелизације резултата. Дизајнирана је да омогући флексибилно тестирање физичких сценарија кроз јасно раздвајање конфигурације, симулације и постпроцесирања. Ток извршавања функције `main` приказан је дијаграмом на Слици 6.3.

Дизајн функције `main` омогућава једноставно проширење новим сценаријима и функционалностима. Као део ове функције, шаблонска функција `save_to_vtk` омогућава извоз релевантних поља у стандардном VTK (енг. *Visualization Toolkit*) [46] формату, компатибилном са алатима за визуелизацију попут ParaView [47], VisIt [48] или Mayavi [49]. Слободна површина флуида дефинисана је нултим нивоом функције ниво скупа `fluidSDF`. За екстракцију површинске мреже примењен је контурни филтер из VTK библиотеке који имплементира алгоритам *марширајућих коцки* (енг. *marching cubes*).



Слика 6.3: Дијаграм тока извршавања функције main.

Резултати који садрже искључиво слободну површину извозе се по временском кораку у .obj формату, док се комплетни подаци (притисак, брзина, вртложност) опционо могу експортирати у .vtk формату.

Глава 7

Резултати симулација

У овој глави приказани су резултати симулација извршених кодом за симулацију невискозних и нестипљивих флуида са слободним површинама, који је развијен на темељу ковекторског формализма. Анализирана су два репрезентативна примера вртложних кретања, кључних за разумевање турбуленције и стабилности струјања: *Тејлор-Гринов вртлоџ* (енг. *Taylor–Green vortex*) и *торусни вртлоџ* (енг. *torus vortex*). Први илуструје како нелинеарност Навије–Стоксових једначина из једноставног почетног стања генерише сложену динамику, док тороидалне и полоидалне варијанте торусног вртлога показују утицај геометријске и тополошке структуре на понашање у идеалном флуиду. Коначно, представљени су и тест-примери: пад водене сфере у базен и извори флуида на суседним вертикалним зидовима. Све симулације су извршене на рачунару са процесором AMD RYZEN 7 5700X и 64 GB радне меморије, у прорачунским доменима облика правоугаоних паралелопипеда различитих резолуција.

7.1 Тејлор–Гринов вртлог

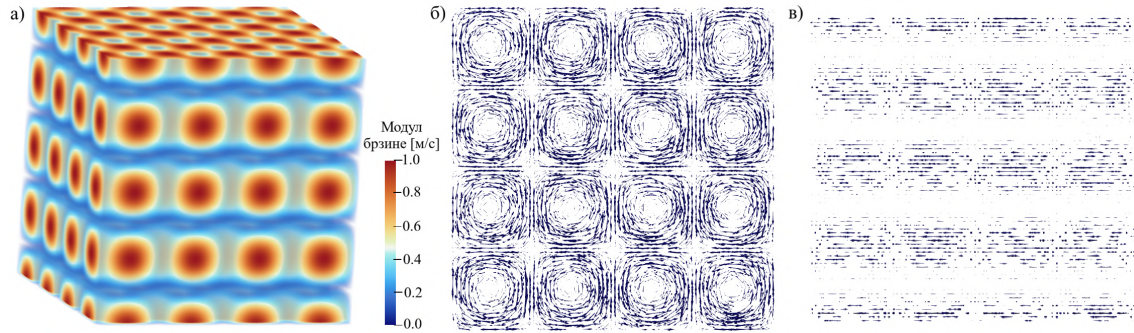
Тејлор-Гринов вртлог је специфичан пример периодичног вртложног тока за који постоји аналитичко решење Навије–Стоксових једначина (2.1) [50]. Прорачунски домен је иницијално подељен на два региона помоћу функције ниво скупа: флуид испуњава област $y < 0.8 \cdot L_y$ (где је $L_y = N_y \cdot \Delta y$ висина домена), док је остатак домена дефинисан као ваздух ($\phi = +5$). Поље брзине се иницијализује аналитичким изразима који одговарају тродимензионалној

Тејлор-Гриновој конфигурацији са подесивим интензитетом Γ :

$$\begin{aligned} u(x, y, z) &= \Gamma \cdot \sin(x) \cdot \cos(y) \cdot \cos(z) \\ v(x, y, z) &= 0 \\ w(x, y, z) &= -\Gamma \cdot \cos(x) \cdot \cos(y) \cdot \sin(z) \end{aligned} \quad (7.1)$$

где су координате $x = i \cdot \Delta x$, $y = j \cdot \Delta y$, $z = k \cdot \Delta z$ дефинисане на чворовима основне мреже. Овај избор компоненти осигурава да је поље брзине недивергентно ($\nabla \cdot \mathbf{u} = 0$) унутар флуидног региона, јер се парцијални изводи међусобно поништавају.

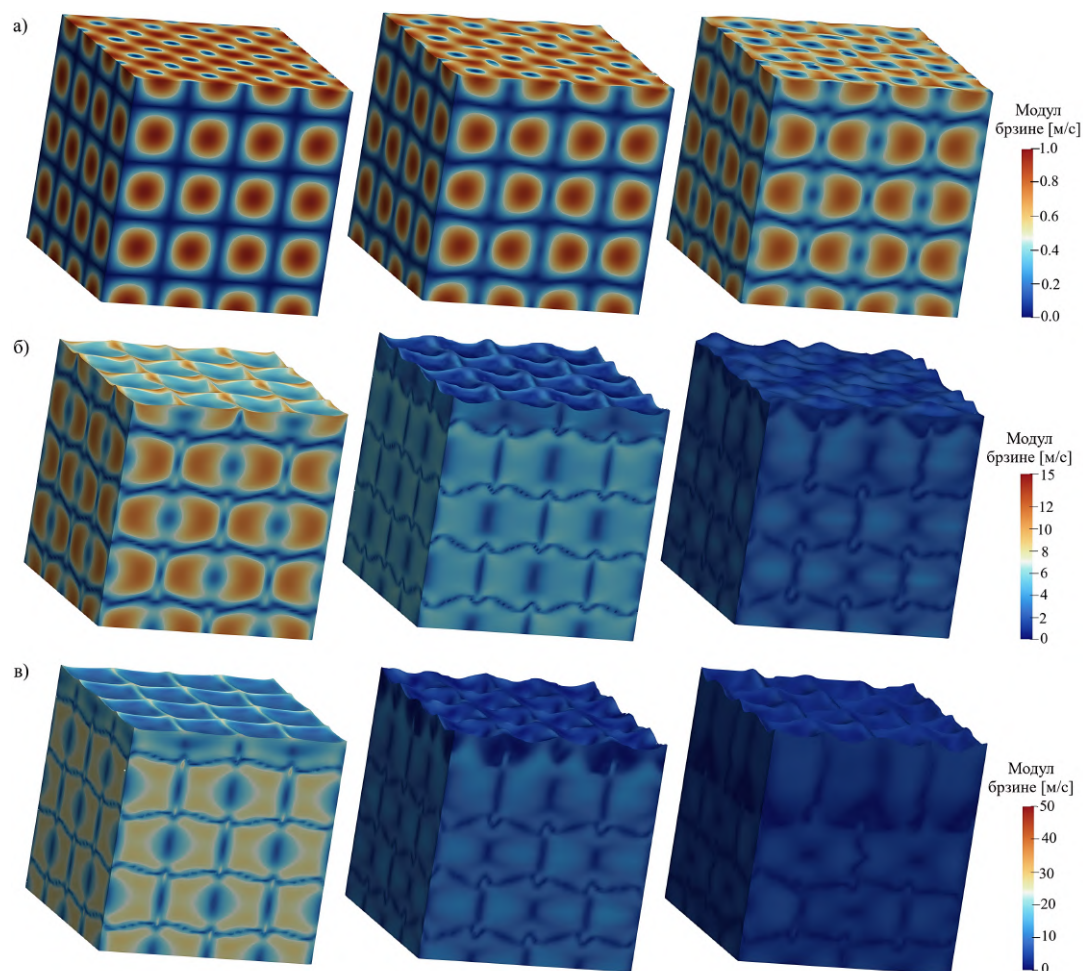
Имплементација у коду користи петље `forEachU` и `forEachW` за постављање u и w компоненти на одговарајућим страницама МАС мреже, док се v компонента имплицитно задржава на нули. Услов `if (j * dy < 0.8 * Domen[1] * dy)` осигурава да се брзина поставља само унутар флуидног региона, спречавајући нумеричке артефакте на граници слободне површине. Нумеричко решавање је неопходно јер нелинеарност Навије-Стоксових једначина из глатког почетног стања генерише сложену динамику типичну за рани стадијум турбулентности.



Слика 7.1: Почетне брзине за Тејлор-Гринов вртлог: а) модул брзине, б) вектори брзине у x - z равни и в) вектори брзине у x - y равни. Резолуција домена: $80 \times 100 \times 80$.

На Слици 7.1 приказане су иницијалне вредности брзина за Тејлор-Гринов вртлог, дефинисане релацијом (7.1). С обзиром на то да ова релација представља егзактно решење само у случају кубних области, флуид испуњава прорачунски домен само делимично. Изнад флуида је зона испуњена ваздухом, чиме је уведена слободна површина. Због њеног присуства, релација (7.1) не представља егзактно решење, већ искључиво почетни услов. Основна

предност овако формулисаних почетних услова јесте то што су недивергентни, чиме је од самог почетка задовољен закон одржања масе. Тиме је обезбеђена физичка веродостојност симулације, пошто дивергентна поља брзине воде ка нефизичким решењима.



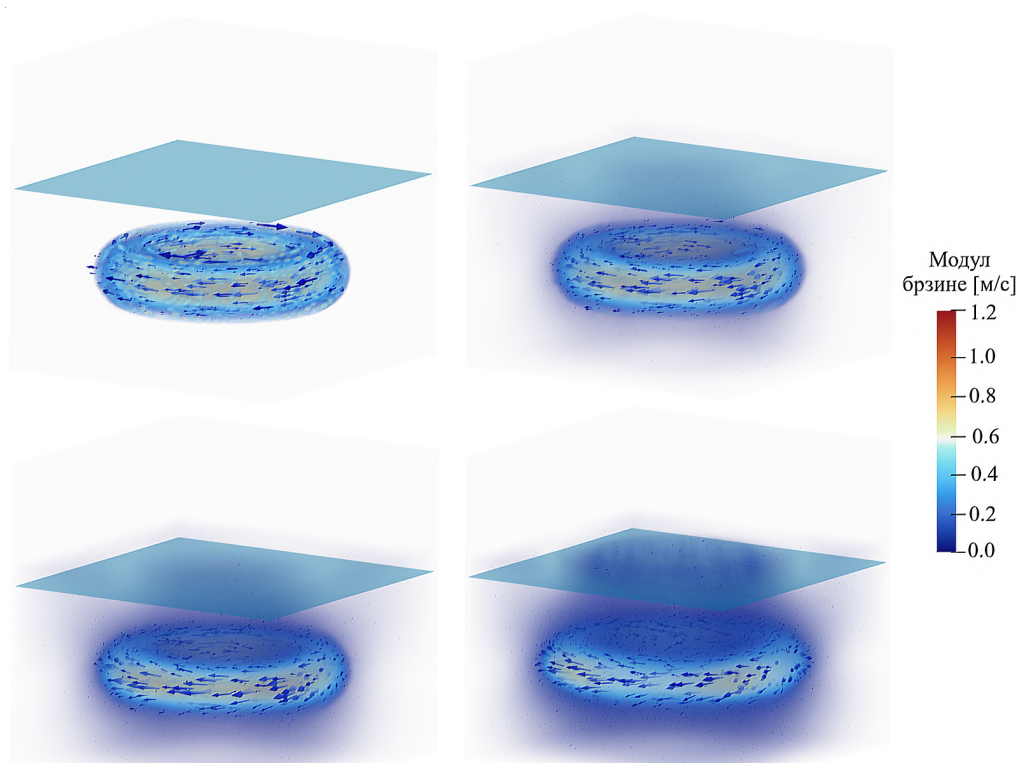
Слика 7.2: Временска еволуција модула брзине за Тејлор-Гринов вртлог за: а) $\Gamma = 1$, б) $\Gamma = 15$ и в) $\Gamma = 50$. Време извршавања је 1700 s по симулацији.

Временска еволуција модула брзине (Слика 7.2) при различитим вредностима параметра Γ показује да његово повећање условљава брже одступање нумеричког решења од аналитичког, изведеног за област без слободне површине. Ово одступање је директна последица нарастајућих деформација слободне површине, које у систем уносе изражене нелинеарне ефекте и појачану дисипацију. Валидност симулације потврђена је очувањем запремине флуида у времену.

7.2 Торусни вртлог

Овде ћемо приказати резултате симулација за два типа торусних вртлога: *тороидални* и *полоидални* вртлог. Ова терминологија је у складу са дефиницијама из литературе о динамици флуида, премда не постоји потпуна терминолошка уједначеност [51]. Код тороидалног вртложног прстена поље брзина циркулише у хоризонталним равнинама (нормалним на вертикалну осу симетрије торуса). У случају полоидалног вртложног прстена брзина циркулише у меридијалним равнинама (нормалним на кружницу која пролази кроз центар торуса и вертикалну осу), што индукује самопропулзију дуж осе симетрије у складу са класичном теоријом вртложних прстенова [7, 51].

Тороидални вртложни прстен



Слика 7.3: Временска еволуција брзине флуида и слободне површине тороидалног вртложног прстена, за временске тренутке (редом): 0 s, 5 s, 10 s и 20 s. Осим модула приказани су и вектори брзина. Резолуција домена: $80 \times 80 \times 80$, време извршавања симулације: 680 s.

За симулацију вртложних структура имплементиран је торусни вртлог чија брзина циркулише у хоризонталној $x-z$ равни око вертикалне y -осе симетрије. Ова конфигурација, коју називамо тороидалним вртлогом (према правцу поља брзине), генерише вртложност оријентисану у меридијалним правнима, које садрже осу симетрије. Геометрија торуса је дефинисана великим полупречником $R = 0.3 \cdot L_x$ који представља растојање од центра торуса до средишта прстена, и малим полупречником $r = 0.1 \cdot L_x$ који одређује дебљину попречног пресека. Центар торуса је позициониран у тачки:

$$\mathbf{c} = \left(\frac{L_x}{2}, 0.3 \cdot L_y, \frac{L_z}{2} \right), \quad (7.2)$$

где су $L_x = N_x \cdot \Delta x$, $L_y = N_y \cdot \Delta y$, $L_z = N_z \cdot \Delta z$ димензије области у којој се врши прорачун. Иницијализација поља брзине врши се у три корака за сваку тачку $\mathbf{x} = (x, y, z)$ у флуиду ($y < 0.6 \cdot L_y$):

- *Корак 1* - Пројекција на централни круг торуса: за дату тачку, израчунава се радијално растојање у хоризонталној равни:

$$\rho = \sqrt{(x - c_x)^2 + (z - c_z)^2}, \quad (7.3)$$

а затим пројекција на прстен полупречника R :

$$\mathbf{x}_{\text{proj}} = \left(c_x + R \cdot \frac{x - c_x}{\rho}, c_y, c_z + R \cdot \frac{z - c_z}{\rho} \right). \quad (7.4)$$

- *Корак 2* - Израчунава се растојање од језгра вртлога:

$$d = \|\mathbf{x} - \mathbf{x}_{\text{proj}}\| = \sqrt{(x - x_{\text{proj}})^2 + (y - c_y)^2 + (z - z_{\text{proj}})^2}. \quad (7.5)$$

- *Корак 3* - Израчунава се поље брзина са линеарним профилем. Тангентни вектор у $x-z$ равни (ротација за 90°) дат је као:

$$\mathbf{t} = \frac{1}{\sqrt{(x - c_x)^2 + (z - c_z)^2}} \begin{pmatrix} -(z - c_z) \\ 0 \\ (x - c_x) \end{pmatrix} \quad (7.6)$$

Интензитет брзине са линеарним опадањем унутар језгра је:

$$v(d) = \begin{cases} \frac{\Gamma}{2\pi R} \cdot \frac{r - d}{r}, & d < r \quad (\text{унутар језгра}) \\ \frac{\Gamma}{2\pi R} \cdot \max\left(0, \frac{r + \Delta x/2 - d}{\Delta x/2}\right), & d \geq r \quad (\text{споља}) \end{cases} \quad (7.7)$$

где је $\Gamma = 10$ циркулација, а Δx просторни корак мреже. Коначно поље брзина израчунава се као: $\mathbf{u}(\mathbf{x}) = v(d) \cdot \mathbf{t} = \begin{pmatrix} u \\ 0 \\ w \end{pmatrix}$.

Оваква конфигурација генерише стационарну вртложну структуру без кретања дуж осе симетрије, што је чини погодном за тестирање очувања енергије и вортицитета при дуготрајним симулацијама.

Слика 7.3 илуструје временску еволуцију поља брзине и слободне површине за тороидални прстен. Вртложност се споро губи током времена, претежно остајући у $x-z$ равни, док се промене брзине релативно споро шире кроз запремину флуида. Стога деформације слободне површине остају занемарљиве, у складу са теоријским предвиђањима за овај тип вртлога, чиме је потврђена коректност симулације [51].

Полоидални вртложни прстен

За разлику од тороидалног случаја, код овог типа вртлога брзинско поље циркулише у меридијалним равнима (равнима које садрже осу симетрије y). Ова конфигурација индукује самопропулзију прстена дуж осе симетрије, представљајући строг тест очувања импулса и кохерентности вртложне структуре. Геометријски параметри су идентични као код тороидалног случаја ($R = 0.3 \cdot L_x$, $r = 0.1 \cdot L_x$), али је физичка природа поља брзине другачија. Иницијализација поља брзине заснива се на *Ранкиновом моделу*, који комбинује чврсту ротацију унутар језгра и потенцијални ток споља [8]. За сваку тачку $\mathbf{x} = (x, y, z)$ у флуидном домену израчунава се:

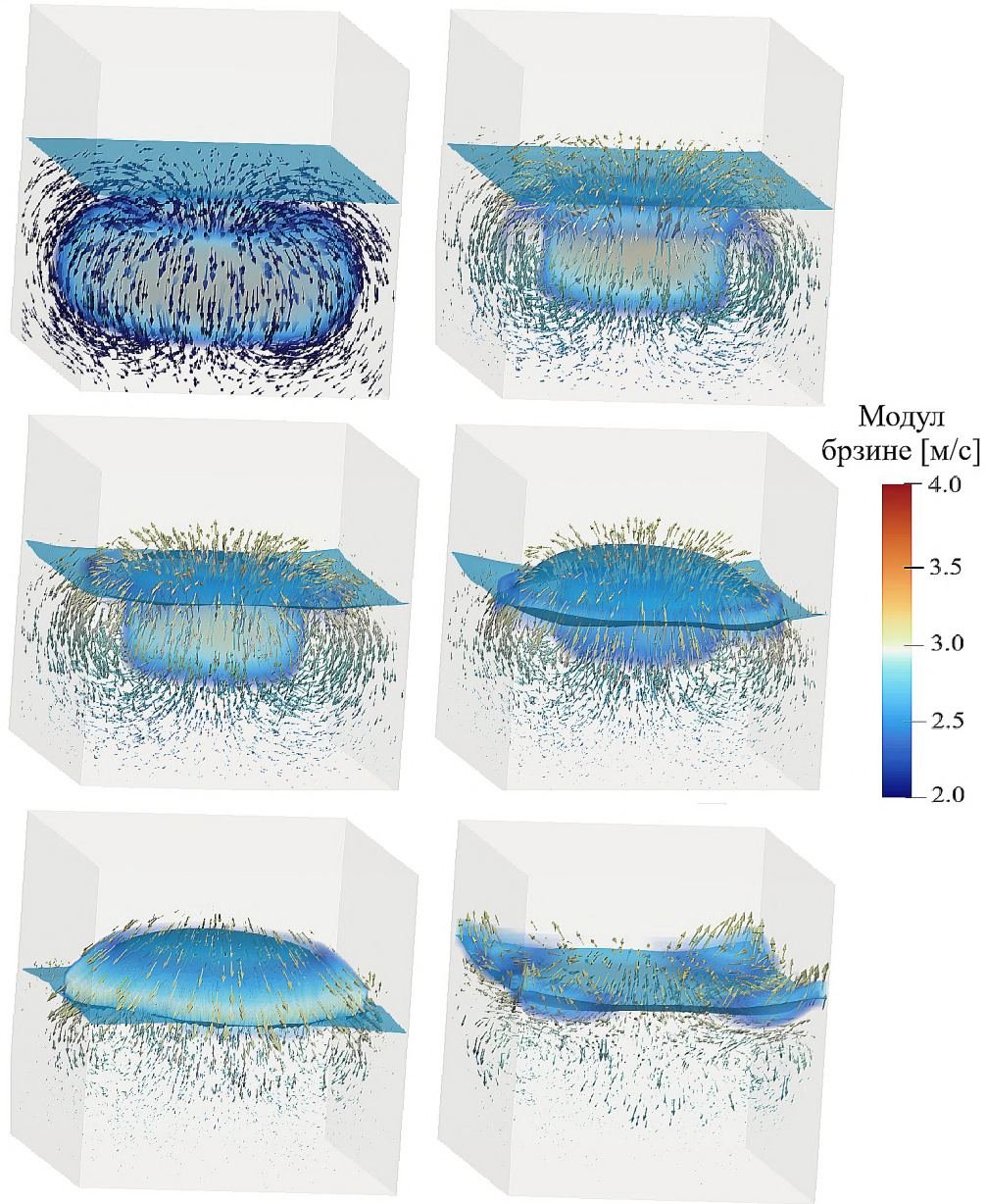
- *Корак 1* - Координате у меридијалној равни: радијално растојање од осе симетрије (y -осе): $\rho = \sqrt{(x - c_x)^2 + (z - c_z)^2}$. Растојање од централне линије прстена у меридијалној ρ - y равни:

$$\Delta\rho = \rho - R, \quad \Delta y = y - c_y, \quad d = \sqrt{\Delta\rho^2 + \Delta y^2}. \quad (7.8)$$

- *Корак 2* - Ранкинов профил брзине:

$$v_\theta(d) = \begin{cases} \frac{\Gamma \cdot d}{2\pi r^2}, & d \leq r \quad (\text{чврста ротација унутар језгра}) \\ \frac{\Gamma}{2\pi d}, & d > r \quad (\text{потенцијални ток споља}) \end{cases} \quad (7.9)$$

где је $\Gamma = 10$ циркулација, а r полупречник језгра вртлога.



Слика 7.4: Временска еволуција брзине флуида и слободне површине за по-лоидални вртложни прстен, за временске тренутке (редом): 0 s, 5 s, 8 s, 12 s, 20 s и 30 s. Осим модула приказани су и вектори брзина. Резолуција домена: $80 \times 80 \times 80$, време извршавања симулације: 820 s.

- *Корак 3* - Тангентни вектор у меридијалној равни: смер циркулације се контролише параметром $\sigma = \pm 1$: $\mathbf{t}_{\rho y} = \frac{\sigma}{d} \begin{pmatrix} -\Delta y \\ \Delta \rho \end{pmatrix}$, где $\sigma = +1$ даје кретање нагоре ($+y$), а $\sigma = -1$ надоле ($-y$).

- *Корак 4* - Пресликавање у 3D Декартове компоненте:

$$\mathbf{u}(\mathbf{x}) = v_\theta(d) \cdot \begin{pmatrix} t_\rho \cdot \frac{x - c_x}{\rho} \\ t_y \\ t_\rho \cdot \frac{z - c_z}{\rho} \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (7.10)$$

где се компонента t_ρ скалира са $(x - c_x)/\rho$ и $(z - c_z)/\rho$ да би се добила правилна оријентација у 3D простору. За тачке на самој оси симетрије ($\rho < 10^{-8}$), брзина се поставља на нулу да би се избегао сингуларитет координата.

Према класичној теорији танких вртложних прстенова [7], оваква конфигурација индукује сопствено кретање дуж осе симетрије брзином:

$$U_{\text{self}} = \frac{\Gamma}{4\pi R} \left[\ln \left(\frac{8R}{r} \right) - \frac{1}{4} \right] + O \left(\frac{r^2}{R^2} \right) \quad (7.11)$$

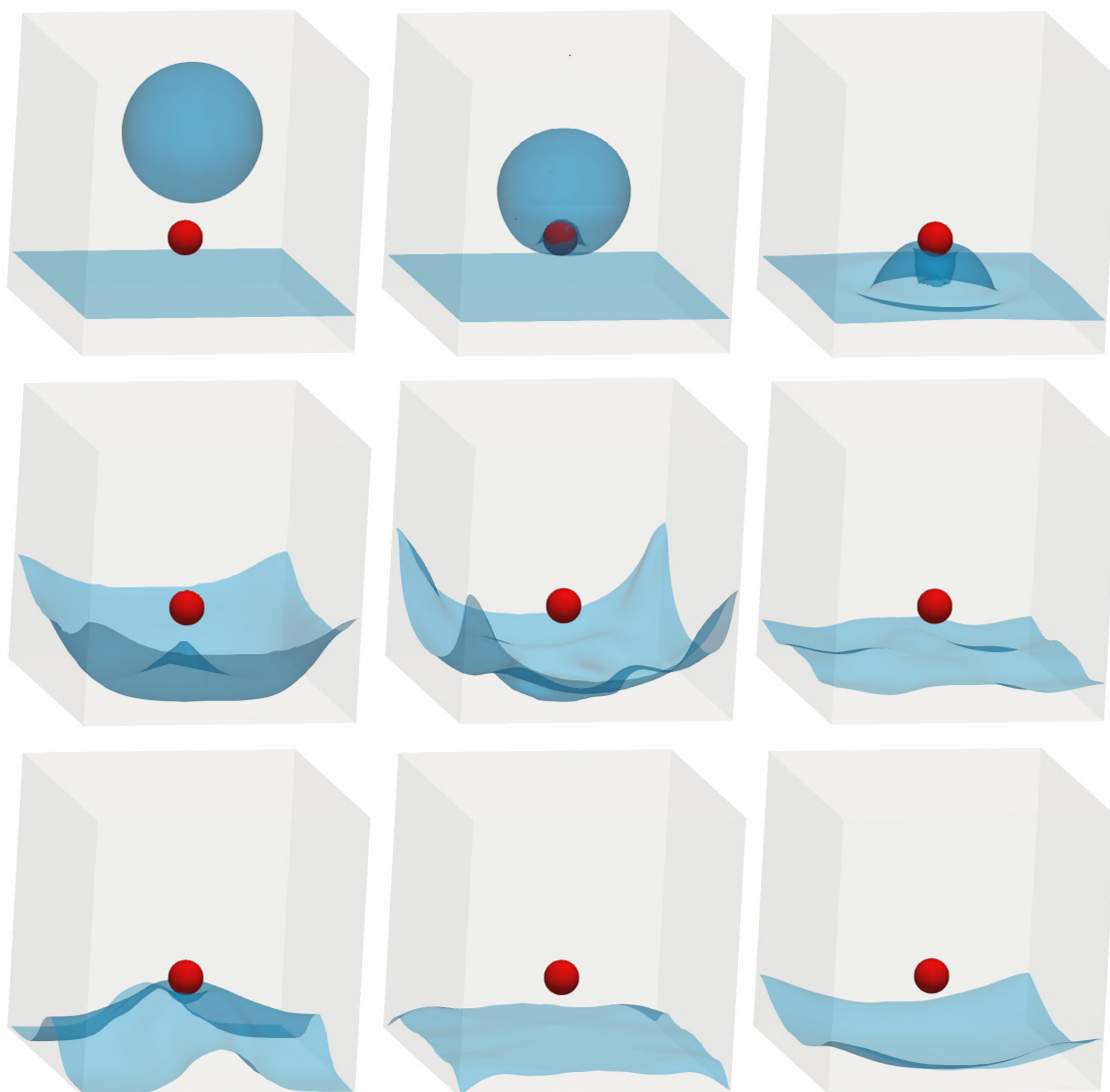
Ову брзину називамо *самоиндукованом брзином кретања вртложа*. За типичне параметре у овој имплементацији ($\Gamma = 10$, $R = 0.3 \cdot L_x$, $r = 0.1 \cdot L_x$), теоријска брзина кретања износи приближно $U_{\text{self}} \approx 2.5$ m/s.

Слика 7.4 приказује временску еволуцију поља брзине и слободне површине за полоидални вртложни прстен. За разлику од стационарног тороидалног случаја, овај вртлог карактерише самопропулзија, односно индуковано кретање дуж вертикалне осе симетрије. За почетне услове са Сlike 7.1a, прстен се креће нагоре, ка слободној површини. Оваква динамика изазива израженије деформације слободне површине и интензивнију дисипацију вртложности. Уочена самопропулзија и усмерено кретање ка слободној површини представљају физички очекивано понашање, чиме се потврђује реалистичност симулације.

7.3 Пад водене сфере у базен

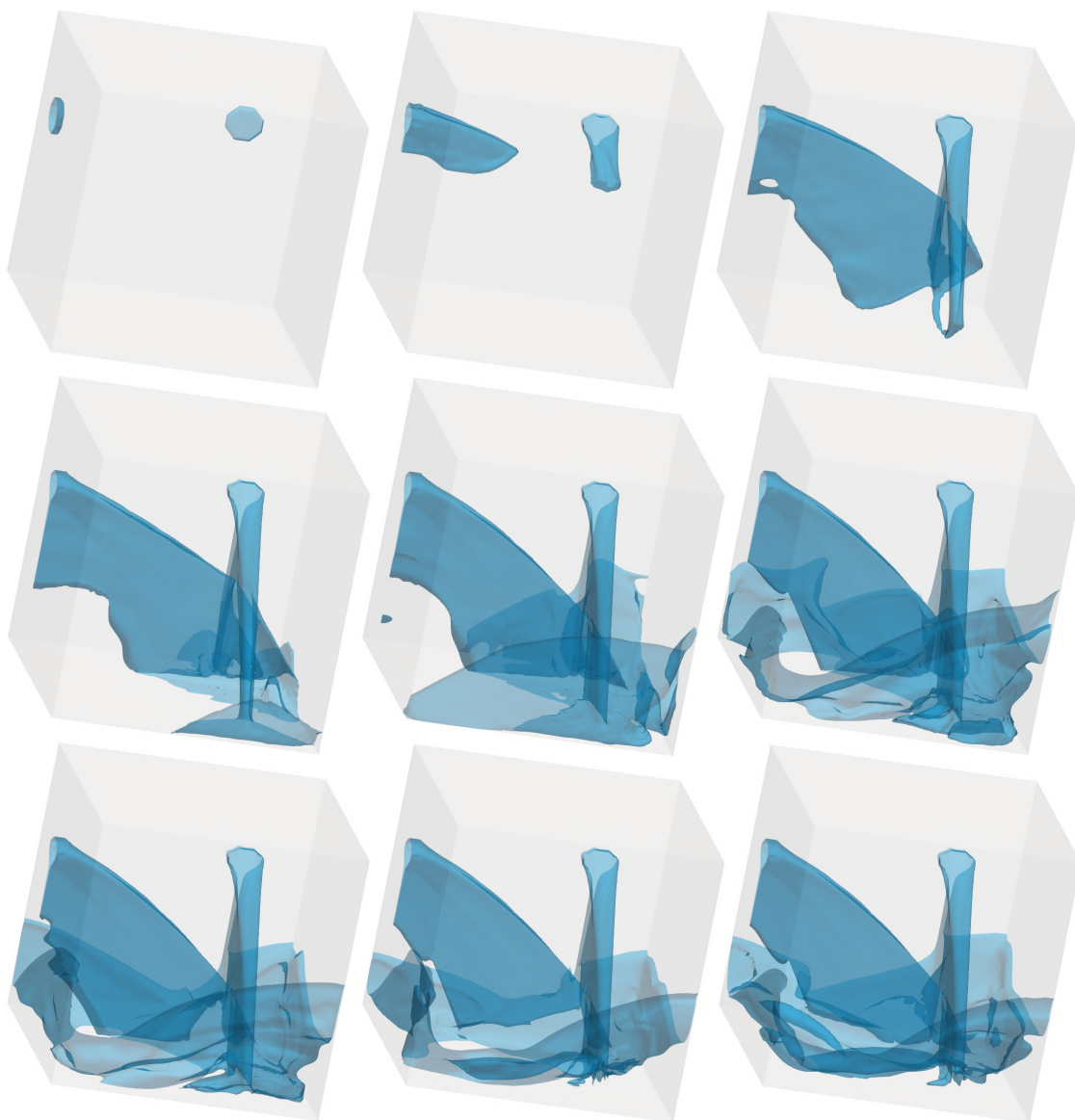
Слика 7.5 приказује резултате симулације пада водене сфере у базен, при чему она наилази на малу чврсту сферну препреку. Иницијализација је поједностављена јер кретање потиче искључиво од гравитације, чиме је почетно поље брзине недивергентно. Због велике димензије водене сфере, занемаривање површинског напона не уноси значајну грешку, што не би био случај код

кретања водених капи. Иако струјање није доминантно вртложно, локална вртложност се јавља услед интеракције флуида са границама прорачунског домена. Валидност нумеричког поступка потврђена је адекватном репродукцијом наведених појава, укључујући спајање флуидних области.



Слика 7.5: Пад водене сфере у базен за временске тренутке (редом): 0 s, 0.6 s, 1.2 s, 1.8 s, 2.4 s, 3.0 s, 3.6 s, 4.2 s и 5 s. Резолуција домена: $80 \times 80 \times 80$, време извршавања симулације: 1400 s.

7.4 Извори флуида на суседним вертикалним зидовима



Слика 7.6: Истицање флуида из извора на суседним вертикалним зидовима, за временске тренутке (редом): 0 s, 0.4 s, 1 s, 1.5 s, 2 s, 4 s, 6 s, 10 s и 20 s. Резолуција прорачунске области: $90 \times 90 \times 90$, време извршавања симулације: 4300 s.

На Слици 7.6 приказани су резултати симулације флуида који истиче константном хоризонталном брзином из извора на суседним вертикалним зидовима правоугаоне паралелопипедне области, формирајући вртлог у смеру казаљке на сату (посматрано одозго). Иако се овај сценарио може реализовати помоћу одговарајућих улазних граничних услова (одељак 2.4), због потешкоћа са тим приступом примењен је алтернативни поступак. Извори су моделовани као танки цилиндри смештени непосредно уз зидове, унутар којих се у сваком временском кораку обнавља константна хоризонтална брзина усмерена ка унутрашњости домена. Иницијална слободна површина одређена је истовремено са функцијом ниво скупа која описује цилиндри. Реалистичност симулације потврђена је формирањем стабилног вртлога у очекиваном смеру, што одговара теоријским предвиђањима за овакву конфигурацију извора.

Глава 8

Закључак

Развој реалистичних симулација флуида представља један од централних изазова савремене рачунарске графике. У овом мастер раду представљен је и нумерички имплементиран ковекторски приступ за симулацију кретања флуида са слободним површинама. Полазећи од ограничења традиционалних Ојлерових шема, у којима семи-Лагранжова адвекција поља брзине уноси нумеричку дифузију и нарушава Келвинову теорему о очувању циркулације, коришћен је геометријски конзистентан ковекторски формализам [11]. Брзина флуида формулисана је као диференцијална 1-форма која се адвектује применом Лијевог извода, чиме се обезбеђује задовољење фундаменталних закона очувања вртложности, док се истовремено елиминишу главни извори нумеричке дисипације.

За потребе овог мастер рада развијен је тродимензионалан кôд имплементиран у модуларној C++ архитектури и помереној MAC мрежи, са интегрисаном методом ниво скупа за прецизно праћење слободних површина и периодичном реиницијализацијом функције означеног растојања. Валидација модела је извршена на низу тродимензионалних тест случајева који обухватају Тејлор–Гринов вртлог, торусне вртлоге, пад водене сфере у базен и истицање флуида из извора на суседним вертикалним зидовима. Добијени резултати потврђују да ковекторски приступ омогућава стабилне и физички конзистентне симулације нестишљивог и невискозног флуида, чиме се постиже значајан напредак у нумеричком моделовању струјања. Посебно је значајна могућност очувања вртложних структура током дуготрајних симулација, чиме се омогућавају визуелно уверљиве анимације у рачунарској графици.

У поређењу са индустријским стандардима попут FLIP методе [19, 22], ковекторски приступ нуди ефикасно очување вртложних структура без потребе за експлицитним праћењем честица. Иако FLIP метода такође смањује нумеричку дисипацију, она захтева значајно већи број честица за постизање сличног квалитета, што повећава рачунске захтеве. Ковекторска метода, с друге стране, задржава ефикасност Ојлеровог приступа уз геометријски конзистентно очување циркулације.

Имплементирани алгоритам показао се робусним, ефикасним и погодним за примене у рачунарској графици где се захтева висок степен визуелне реалистичности уз очување физичке веродостојности. Паралелизација применом OpenMP директива, увођење адаптивних временских поткорака ради задовољења CFL услова и оптимизовано управљање меморијом обезбедили су скалабилност кода чак и на мрежама више резолуције. Компатибилност са стандардним VTK и OBJ форматима омогућава директно коришћење симулационих података у постојећим алатима за визуелизацију и анимацију.

На основу добијених резултата може се закључити да ковекторска метода представља практичан алат за развој *геометријских нумеричких алгоритама* (енг. *geometric structure-preserving algorithms*), који представљају тренутно најсофистициранији приступ за симулацију флуида. Ковекторска метода не само да ефикасно решава проблем нумеричке дисипације вртложности, већ и отвара пут ка развоју нових генерација симулатора који се темеље на принципима диференцијалне геометрије и топологије.

Према сазнањима аутора, развијени код представља прву имплементацију ковекторске методе за симулацију струјања флуида са слободним површинама засновану на Ојлеровом приступу. Будућа истраживања се могу усмерити ка проширењу методе на вискозне и стишљиве флуиде, ка адаптивном рафинисању мреже у близини граничне површине, као и ка имплементацији на специјализованом хардверу за интерактивне апликације у реалном времену.

Библиографија

- [1] J. Bender, D. Koschier, T. Charrier, and D. Brand, “Position-based simulation methods in computer graphics”, *Computer Graphics Forum*, **33**, 252–266, (2014).
- [2] R. Bridson, *Fluid simulation for computer graphics*, 2nd ed. (CRC Press, 2016).
- [3] S. Zhang, Y. Liu, and J. Chang, “Physics-based fluid simulation in computer graphics”, *Computational Visual Media*, **10**, 195–233, (2024).
- [4] D. Holz, S.R. Jeske, F. Löffner, J. Bender, Y. Yang, and S. Andrews, “Multiphysics Simulation Methods in Computer Graphics”, *Computer Graphics Forum*, **44**, e70082(1–39), (2025).
- [5] J.H. Ferziger and M. Perić, *Computational methods for fluid dynamics*. (Springer Science & Business Media, 2012).
- [6] X. Wang, Y. Xu, S. Liu, B. Ren, J. Kosinka, A.C. Telea, J. Wang, C. Song, J. Chang, C. Li, J.J. Zhang, and X. Ban, “Physics-based fluid simulation in computer graphics: Survey, research trends, and challenges”, *Comp. Visual Media*, **10**, 803–858, (2024).
- [7] G.K. Batchelor, *An Introduction to Fluid Dynamics*. (Cambridge University Press, 2012).
- [8] A.J. Majda and A.L. Bertozzi, *Vorticity and incompressible flow*. (Cambridge University Press, 2002).
- [9] A.K. Košir and H.G. Tomc, “Visual effects and their importance in the field of visual media creation”, *Journal of Graphic Engineering and Design*, **13**, 5–13 (2022).

- [10] N.H. Lu, “Magic of Water: Exploration of Production Process with Fluid Effects in Film and Advertisement in Computer-Aided Design”, *Eng. Proc.*, **98**, 20(1–9), (2025).
- [11] M.S. Nabizadeh, S. Wang, R. Ramamoorthi, and A. Chern, “Covector Fluids”, *ACM Trans. Graph.*, **41**, 113(1–15), (2022).
- [12] J.A. Sethian, *Level set methods and fast marching methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. (Cambridge University Press, 1999).
- [13] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. (Springer-Verlag New York, Inc., 2003).
- [14] L.D. Landau and E.M. Lifshitz, *Fluid Mechanics*. (Pergamon Press, 1987).
- [15] The Millennium Prize Problems, доступно на: <https://www.claymath.org/millennium-problems/>.
- [16] D. Prodanović, *MEHANIKA FLUIDA za studente Gradjevinskog fakulteta*. (Gradjevinski fakultet, Univerzitet u Beogradu i Akademska misao, 2024).
- [17] C. Kimberling, “Emmy Noether”, *American Mathematical Monthly*, **79**, 136–149, (1972).
- [18] Mantaflow, доступно на: <http://mantaflow.com/>.
- [19] Fluid Engine Dev (FED), доступно на: <https://github.com/doyubkim/fluid-engine-dev>.
- [20] SPliHSplash, доступно на: <https://splishsplash.physics-simulation.org/>.
- [21] Taichi, доступно на: <https://www.taichi-lang.org/>.
- [22] SideFX Houdini, доступно на: <https://www.sidefx.com/>.
- [23] Next Limit RealFlow, доступно на: <https://realflow.com/>.
- [24] Chaos Phoenix, доступно на: <https://www.chaos.com/phoenix>.
- [25] Autodesk Bifrost, доступно на: <https://www.autodesk.com/products/maya/overview>.

- [26] JangaFX EmberGen, доступно на: <https://jangafx.com/software/embergen>.
- [27] Obi Fluid, доступно на: <https://obi.virtualmethodstudio.com/>.
- [28] D. Kim, *Fluid Engine Development*. (CRC Press, 2017).
- [29] F. Harlow and J. Welch, “Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface”, *Phys. Fluids*, **8**, 2182–2189, (1965).
- [30] H.F. Trotter, “On the product of semi-groups of operators”, *Proceedings of the American Mathematical Society*, **10**, 545–551, (1959).
- [31] G. Strang, “On the Construction and Comparison of Difference Schemes”, *SIAM Journal on Numerical Analysis*, **5**, 506–517, (1968).
- [32] J. Stam, “Stable Fluids”, in *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pp. 121–128, (New York: ACM, 1999).
- [33] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, 1992).
- [34] M. Garcia-Villalba, T. Colonius, O. Desjardins, D. Lucas, A. Mani, D. Marchisio, O.K. Matar, F. Picano, and S. Zaleski, “Numerical methods for multiphase flows”, *International Journal of Multiphase Flow*, **191**, 105285 (1–27), (2025).
- [35] D. Rossi, S. Di Giorgio, and S. Pirozzoli, “Comparative analysis of volume of fluid and phase-field methods for numerical simulations of two-phase flows”, *International Journal of Multiphase Flow*, **189**, 105245(1–26), (2025).
- [36] J.A. Sethian and P. Smereka, “Level set methods for fluid interfaces”, *Annu. Rev. Fluid Mech.*, **35**, 341–372, (2003).
- [37] K. Khedkar, A.C. Mamaghani, P. Ghysels, N.A. Patankar, and A.P.S. Bhalla, “Preventing mass loss in the standard level set method - New insights from variational analyses”, *Journal of Computational Physics*, **520**, 113495(1–40), (2025).

- [38] R. Goldade, C. Batty, and C. Wojtan, “A Practical Method for High-Resolution Embedded Liquid Surfaces”, *Computer Graphics Forum*, **35**, 233–242 (2016).
- [39] J.L. Lagrange, *Mécanique Analitique*. (Ve Courcier, 1811).
- [40] V.I. Arnold and B.A. Khesin, *Topological Methods in Hydrodynamics*. (Springer Science & Business Media, 1999).
- [41] J.E. Marsden and T.S. Ratiu, *Introduction To Mechanics And Symmetry - A Basic Exposition of Classical Mechanical Systems*, 2nd ed. (Springer New York, 1999).
- [42] F. Feng, J. Liu, S. Xiong, S. Yang, Y. Zhang, and B. Zhu, “Impulse Fluid Simulation”, *IEEE Trans Vis Comput Graph.*, **28**, 3081–3092, (2023).
- [43] S. Wang, M.S. Nabizadeh, and A. Chern, *Exterior Calculus in Graphics*. Course notes for a SIGGRAPH 2023 course, доступно на: <https://stephaniewang.page/ExteriorCalculusInGraphics/>.
- [44] D. Milinković, *Uvod u račun na mnogostrukostima*, доступно на: www.matf.bg.ac.rs/~milinko.
- [45] V.A. Zorich, *Mathematical Analysis II*, 2nd ed. (Springer Berlin, 2016).
- [46] VTK, доступно на: <https://vtk.org/>.
- [47] ParaView, доступно на: <https://www.paraview.org/>.
- [48] VisIt, доступно на: <https://visit-dav.github.io/visit-website/index.html>.
- [49] Mayavi, доступно на: <https://github.com/enthought/mayavi>.
- [50] M. Antuono, “Tri-periodic fully three-dimensional analytic solutions for the Navier–Stokes equations”, *Journal of Fluid Mechanics*, **890**, A23(1–15), (2020).
- [51] P.G. Saffman, *Vortex Dynamics*. (Cambridge University Press, 2012).