

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA JADERNÁ
A FYZIKÁLNĚ INŽENÝRSKÁ**



**BAKALÁŘSKÁ
PRÁCE**

2023

**MATĚJ
POPĎAKUNIK**

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA JADERNÁ
A FYZIKÁLNĚ INŽENÝRSKÁ**

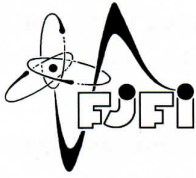
Vysoce přesné algoritmy pro
interpolace funkcí v
Lagrangeovsko-Eulerovských
(ALE) metodách

BAKALÁŘSKÁ PRÁCE

AUTOR PRÁCE: MATĚJ POPĎAKUNIK

VEDOUCÍ PRÁCE: DOC. ING. MILAN KUCHARÍK, PHD.

ŠKOLNÍ ROK: 2022/2023



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA JADERNÁ A FYZIKÁLNĚ INŽENÝRSKÁ
Katedra fyzikální elektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Matěj P o p d' a k u n i k**

Studijní program: **Aplikace přírodních věd**

Obor: **Informatická fyzika**

Akademický rok: **2021/2022**

Název práce: **Vysoce přesné algoritmy pro interpolace funkcí v Lagrangeovsko-Eulerovských (ALE) metodách**
(česky)

Název práce: **Highly-accurate algorithms for function interpolations in arbitrary Lagrangian-Eulerian (ALE) methods**
(anglicky)

Cíl práce:

Při změně výpočetní sítě v ALE simulaci musí nevyhnutelně následovat interpolace dat (remapování) z Lagrangeovské sítě na vyhlazenou. U tradičních metod lze ukázat jejich druhý řád přesnosti v případě hladkých profilů remapovaných veličin, v případě nespojitých dat však chyba významně roste. Cílem práce je prozkoumat možnosti nelineárních (a nepolynomiálních) interpolací dat v rámci kombinované metody používající různé aproximace v jednotlivých buňkách výpočetní sítě.

Pokyny pro vypracování:

1. Seznamte se s metodami pro rekonstrukci a remapování funkcí z diskrétních hodnot [1,2].
2. Implementujte metodu remapu založenou průnicích a přesné integraci [3].
3. Implementujte rekonstrukci pomocí THINC funkce [4,5].
4. Navrhněte a implementujte metodu rekonstrukce pomocí nespojitě funkce.
5. Otestujte navrženou metodu a porovnejte ji s klasickými jednomateriálovými a vícemateriálovými [1] metodami.

Doporučená literatura:

1. Kucharik M., Shashkov M., Conservative multi-material remap for staggered multi-material arbitrary Lagrangian-Eulerian methods, *Journal of Computational Physics* 258, 268-304, 2014.
2. Diot S., Clain S., Loubere R., Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials, *Computers & Fluids* 64, 43-63, 2012.
3. Margolin L.G., Shashkov M., Second-order sign-preserving conservative interpolation (remapping) on general grids, *Journal of Computational Physics* 184, 266-298, 2003.
4. Kucharik M., Loubere R., High-accurate and robust conservative remapping combining polynomial and hyperbolic tangent reconstructions, *Computers & Fluids* 208, 104614, 2020.
5. Xiao F., Honma Y., Kono T., A simple algebraic interface capturing scheme using hyperbolic tangent function, *International Journal for Numerical Methods in Fluids* 48, 1023-1040, 2005.

Jméno a pracoviště vedoucího práce:

doc. Ing. Milan Kuchařík, PhD.

Katedra fyzikální elektroniky, Fakulta jaderná a fyzikálně inženýrská ČVUT v Praze.

Datum zadání bakalářské práce: 15. říjen 2021

Termín odevzdání bakalářské práce: 7. červenec 2022

Doba platnosti zadání je dva roky od data zadání.



.....
Garant oboru



.....
Vedoucí katedry



.....
Děkan

V Praze dne 15.10.2021

PROHLÁŠENÍ

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškerou použitou literaturu.

V Praze dne 1. 8. 2023


Matěj Popďakunik

Poděkování

Chtěl bych zde poděkovat doc. Ing. Milanu Kuchaříkovi, Ph.D. za věnovaný čas, vstřícný přístup a cenné rady při vedení mé bakalářské práce.

Matěj Popd'akunik

Název práce:

Vysoce přesné algoritmy pro interpolace funkcí v Lagrangeovsko-Eulerovských (ALE) metodách

Autor: Matěj Popd'akunik

Obor: Informatická fyzika

Druh práce: Bakalářská práce

Vedoucí práce: doc. Ing. Milan Kuchařík, PhD.

Abstrakt:

Pro hydrodynamické simulace některých problémů jsou vhodné Lagrangeovsko-Eulerovské (ALE) metody pro jejich přesnost a zároveň robustnost. V této práci se studují nepřímé cell-centered ALE metody pro simulaci hydrodynamiky v jedné dimenzi. Nejvíce se soustředí na jeden krok ALE simulací, remap. Jsou popsány metody pro polynomiální a nepolynomiální rekonstrukce. Z nepolynomiálních rekonstrukcí je popsána metoda THINC a v této práci nově odvozená metoda spočívající v rekonstrukci pomocí nespojitě funkce. Všechny popsané metody, spolu s Lagrangeovským krokem, byly následně implementovány v novém 1D ALE kódu. Pomocí tohoto kódu byla následně provedena sada numerických testů. Výsledky těchto testů byly poté srovnány s odbornou literaturou.

Klíčová slova: ALE metody, hydrodynamika, remap, rekonstrukce funkcí

Title:

Highly-accurate algorithms for function interpolations in arbitrary Lagrangian-Eulerian (ALE) methods

Author: Matěj Popd'akunik

Abstract:

For hydrodynamic simulation of certain problems the arbitrary Lagrangian-Eulerian (ALE) methods are preferred for their accuracy and robustness. In this work, indirect cell-centered arbitrary Lagrangian-Eulerian methods for simulating hydrodynamics in one dimension are studied. It focuses on one step of ALE simulations in particular, remap. Methods for polynomial and non-polynomial reconstruction are described. From non-polynomial reconstructions we describe the THINC method and a new method derived in this work, which is based on reconstruction by a discontinuous function. All of the described methods, together with the Lagrangian step, were afterwards implemented in a new 1D ALE code. This code was then used to perform a set of numerical tests. Results of these tests were then compared with published literature.

Key words: ALE methods, hydrodynamics, remap, function reconstruction

Obsah

1	Úvod	9
2	ALE metody	10
2.1	Geometrie výpočetní sítě	11
3	Eulerovský krok ALE metod	12
3.1	Vyhlazování výpočetní sítě	12
3.2	Interpolace veličin mezi sítěmi	12
3.3	Rekonstrukce veličiny	13
3.3.1	Konstantní rekonstrukce	13
3.3.2	Lineární rekonstrukce	13
3.3.3	Lineární rekonstrukce s limiterem	15
3.3.4	Polynomiální rekonstrukce	17
3.3.5	THINC rekonstrukce	18
3.3.6	Rekonstrukce nespojitou funkcí	20
3.3.7	Rekonstrukce v okolí nespojitosti	23
3.3.8	Okrajové podmínky	24
3.4	Výpočet hodnot na nové sítě	24
3.5	Remap v ALE kódu	25
3.6	Multimateriálový remap	25
4	Lagrangeovský krok ALE metod	27
4.1	Fyzikální popis	27
4.2	Diskretizace prvního řádu	28
4.3	Diskretizace druhého řádu	29
4.4	Velikost časového kroku	29
4.5	Okrajové podmínky	30
4.6	Sledování kontaktu	30
4.7	Detekce komprese	31
5	Implementace	32
5.1	Použité knihovny	32
6	Numerické výsledky - remap	33
6.1	Hladký pohyb sítě	33
6.2	Kvadratický polynom a sinus	34
6.3	Nespojitá funkce	36
6.4	Nespojitá nelineární funkce	38
7	Numerické výsledky - ALE metody	42
7.1	Riemmanovy problémy	42
7.2	Shu-Osherův problém	44

1 Úvod

V této práci se zabýváme Eulerovsko-Lagrangeovskými (ALE) metodami pro řešení 1D hydrodynamiky. Dynamika stlačitelné tekutiny je popsána soustavou Eulerových rovnic. Tato soustava hyperbolických diferenciálních rovnic je zřídka řešitelná analyticky, proto pro její řešení používáme numerické metody. Po diskretizaci jednotlivých veličin soustavu aproximujeme za pomoci konečných diferencí. My se v této práci zabýváme takzvanou cell-centered diskretizací.

Počátky řešení hydrodynamiky za pomoci numerických metod sahají do 40. let 20. století [1]. Jednalo se o takzvané Lagrangeovské metody. Tyto metody byly založené na staggered diskretizaci. Jedno z prvních Lagrangeovských numerických schémat bylo vytvořeno John von Neumannem v Los Alamos National Laboratory v roce 1943.

V případě čistě Lagrangeovských simulací může nastat problém s degenerací výpočetní sítě. Ten se snaží odstranit ALE metody, které degeneraci sítě předchází pomocí jejího vyhlazování a následného remapu, v němž jsou veličiny přeneseny z původní sítě na vyhlazenou. V této práci se tedy nejprve seznámíme s nepřímou ALE metodou a dále se soustředíme na krok remapu, u kterého popíšeme několik metod pro vytváření rekonstrukcí funkcí.

V případě hladkých funkcí lze používat standardní metody založené na lineární [2, 3, 4] nebo polynomiální rekonstrukci [5, 6]. Nespojitosti na rozhraní materiálů řeší multimateriálové metody [7, 8, 9], ty však nejsou dostupné v řadě simulačních kódů. Proto se zabýváme aproximací nespojitostí v jednomateriálovém případě za pomoci nepolynomiálních rekonstrukcí. V [10, 11, 12] je odvozená nepolynomiální rekonstrukce založená na THINC funkci v kontextu metody konečných objemů. V [13] je pak tato metoda použita pro remapování nespojitých funkcí. V této práci jsme navrhli novou metodu pomocí nespojitých funkce, která je jednodušší a numericky robustnější. Tuto metodu jsme spolu s polynomiálními a THINC rekonstrukcemi následně implementovali v nově vyvinutém 1D ALE kódu a získané výsledky jsme porovnali s výsledky z odborné literatury.

2 ALE metody

Podle pohybu výpočetní sítě rozlišujeme dva přístupy pro simulace hydrodynamiky, Lagrangeovské a Eulerovské.

V Eulerovském přístupu používáme statickou výpočetní síť. Síť se tedy nemůže poškodit. Tyto metody ale nejsou vhodné na simulace problémů ve kterých dochází k silným deformacím výpočetní oblasti, například k jejímu stlačení nebo rozpínání.

U Lagrangeovského přístupu se výpočetní síť hýbe společně s tekutinou. Lagrangeovský přístup má oproti Eulerovskému řadu výhod. Buňky sítě si nevyměňují hmotu se svými sousedy a ta tedy zůstává konstantní. Důsledkem tohoto je, že se výpočetní síť sama zhustí blízko rázových vln. Dále, pokud máme rozhraní na hranici dvou buňek, toto rozhraní můžeme jednoduše sledovat. Jeho hlavní nevýhodou je problém, kdy například u výpočtu nestabilit, výpočetní síť může sama sebe protnout a poté se naše simulace pokazí.

U ALE metod kombinujeme oba dva tyto přístupy. Provádíme takzvaný Lagrangeovský krok, kdy řešíme diskretizované Eulerovy rovnice a aktualizujeme námi pozorované veličiny. Poté provádíme Eulerovský krok ve kterém, z důvodu zamezení chyb v simulaci, vyhlazujeme naši výpočetní síť.

Ale metody se dají dělit na přímé a nepřímé. U nepřímých ALE metod máme Lagrangeovu a Eulerovu část výpočtu oddělenou do dvou samostatných kroků. Přímými ALE metodami se v této práci nezabýváme.

Celkově se algoritmus pro nepřímé ALE metody se dá popsat následovně:

1. Inicializace sítě - Vytvoříme výpočetní síť na oblasti kterou máme zadanou z počátečních podmínek.
2. Inicializace veličin - Z počátečních podmínek na síť nanese veličiny které při výpočtu používáme, v našem kódu to je hustota, tlak a rychlost proudění. Navíc z těchto veličin za pomoci stavové rovnice spočítáme vnitřní energii.
3. Lagrangeovský krok - V tomto kroku řešíme diskretizované Eulerovy rovnice, tedy vývoj veličin a výpočetní sítě v čase.
4. Eulerovský krok - V tomto kroku síťku vyhladíme a přeneseme veličiny ze staré sítě na novou. Toto provádíme po námi určeném počtu kroků nebo pokud detekujeme problém s výpočetní sítí.
5. Přejdeme zpátky ke kroku 3 dokud simulace nedoběhne do námi určeného finálního času.
6. Vypíšeme výsledky.

Existují dvě základní diskretizace ALE metod. V cell-centered metodách máme všechny diskretizované veličiny umístěné ve středech buňek, u staggered metod místo středů buňek známe některé z veličin v uzlech.

V této práci se zabýváme výhradně cell-centered metodami.

2.1 Geometrie výpočetní sítě

Úlohu řešíme na výpočetní oblasti G . Tuto oblast nejprve diskretizujeme. V 1D případě rozdělíme oblast G na n buněk ve formě intervalů. Jedna z možností inicializace je ekvidistantní rozložení buněk (tj. buňky mají stejnou délku).

Buňky označíme indexem i , číslováme je od 0 (tj. buňka na levém okraji výpočetní oblasti má index 0, buňka na pravém okraji má index n). Oblast kterou buňka i pokrývá označme jako G_i

Krajní body buňky i nazýváme uzly a označujeme je jako $x_{i-\frac{1}{2}}$ pro levý uzel a $x_{i+\frac{1}{2}}$ pro pravý uzel.

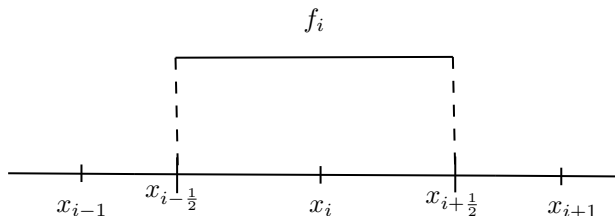
Střed buňky i označujeme jako x_i a definujeme ho následujícím způsobem

$$x_i = \frac{\int_{G_i} x \, dx}{\int_{G_i} 1 \, dx} = \frac{x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}}{2} \quad (1)$$

Objem buňky označíme jako V_i a platí

$$V_i = \int_{G_i} 1 \, dx = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \quad (2)$$

Pro ilustraci značení buněk se podívejme na obrázek 1.



Obrázek 1: Ilustrace značení v buňce i

3 Eulerovský krok ALE metod

Eulerovský krok ALE metod můžeme rozdělit na dvě části. Nejprve provádíme takzvaný rezone, kdy síťku vyhladíme a následně provádíme remap, kdy interpolujeme veličiny a přenášíme je z původní sítě na novou.

Algoritmus pro celý eulerovský krok můžeme rozdělit do následujících 3 kroků.

1. Vytvoření nové sítě, tj. jejích uzlů a spočtení její geometrie.
2. Nalezení interpolační funkce, tomuto kroku budeme říkat rekonstrukce.
3. Výpočet nových hodnot na nové síťce integrací rekonstruované funkce.

3.1 Vyhlazování výpočetní sítě

Při ALE simulaci dochází k vyhlazování sítě. My toto vyhlazení vždy provádíme po určitém počtu kroků, ale existují metody kdy síťku vyhlazujeme pokud zjistíme, že se pokazí. Tento krok označujeme jako rezone.

Laplaceovo vyhlazování

Při Laplaceově vyhlazování je nová souřadnice uzlu určena jako vážený průměr tří uzlů z původní sítě.

$$\hat{x}_{i+\frac{1}{2}} = \frac{x_{i-\frac{1}{2}} + 2x_{i+\frac{1}{2}} + x_{i+\frac{3}{2}}}{4}, \quad (3)$$

kde jsme jako $\hat{x}_{i+\frac{1}{2}}$ označili polohu uzlu $x_{i+\frac{1}{2}}$ na nové síťce. Pro okrajové uzly se poloha nemění, tedy $\hat{x}_{-\frac{1}{2}} = x_{-\frac{1}{2}}$ a $\hat{x}_{n+\frac{1}{2}} = x_{n+\frac{1}{2}}$.

Eulerovské vyhlazování

Při tomto vyhlazování jako nové polohy uzlů nastavíme jejich polohu na začátku simulace.

$$\hat{x}_{i+\frac{1}{2}} = x_{i+\frac{1}{2}}(t = 0) \quad (4)$$

3.2 Interpolace veličin mezi sítěmi

Poté co jsme vyhlazením sítě získali novou síťku, musíme staré veličiny přenést na síťku novou. Tomuto kroku říkáme remap.

Z Lagrangeovského kroku známe hodnoty f_i , které označují střední hodnotu veličin v buňkách na staré síťce. Naším cílem je spočítat nové střední hodnoty na nové síťce. Nejprve tedy provedeme rekonstrukci (interpolaci) průběhu dané veličiny ze starých hodnot a poté pomocí integrace této rekonstruované funkce přes nové buňky spočítáme novou střední hodnotu.

Rekonstrukci v buňce i , kterou nyní budeme hledat, označme jako $f_i(x)$. Poté, co najdeme rekonstrukce $f_i(x)$ ve všech buňkách, budeme pokračovat tím,

že tyto funkce budeme integrovat tak, abychom dostali střední hodnoty veličin na nové síťce.

Pro zjednodušení značení označme

$$f(x) = f_i(x), \forall x \in G_i, \quad (5)$$

tedy $f(x)$ označuje spojení všech rekonstruovaných funkcí.

Požadujeme, aby rekonstrukce byla v každé buňce konzervativní. To znamená, že pro nalezenou lokální interpolační funkci $f_i(x)$ musí platit

$$\frac{\int_{G_i} f_i(x) dx}{V_i} = f_i, \quad (6)$$

kde i označuje index buňky, ve které se nacházíme, G_i označuje oblast, kterou buňka obsahuje, V_i označuje objem buňky (tj. $V_i = \int_{G_i} 1 dx$). Hodnota f_i reprezentuje hodnotu jednotlivých veličin v buňce i , například hustotu ρ_i . Její rekonstrukci bychom označili jako $\rho_i(x)$.

Dále požadujeme konzervativitu celého remapu, tedy

$$\sum_i f_i V_i = \sum_i \hat{f}_i \hat{V}_i, \quad (7)$$

kde \hat{f}_i označuje novou střední hodnotu buňky i a \hat{V}_i označuje její nový objem. Tato podmínka je splněna díky (6).

3.3 Rekonstrukce veličiny

Existují různé možnosti jak rekonstrukci dělat, tyto metody vedou na různé řády přesnosti.

3.3.1 Konstantní rekonstrukce

Nejjednodušší forma rekonstrukce kterou můžeme použít je rekonstrukce konstantní. Pro tuto rekonstrukci předpokládáme že hodnota funkce je v buňce konstantní, tedy

$$f_i(x) = f_i. \quad (8)$$

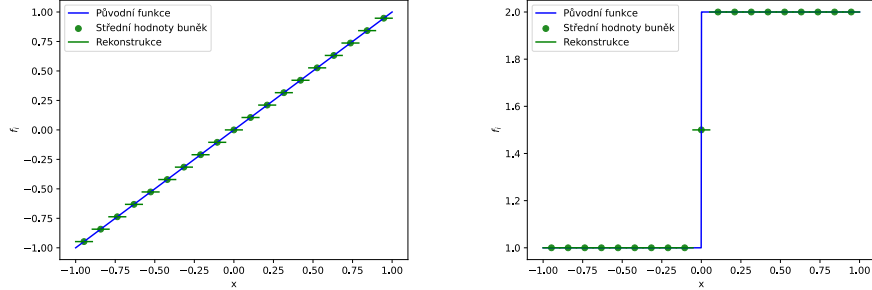
Tato rekonstrukce splňuje podmínku (6) triviálně. Pro ilustraci tuto rekonstrukci můžeme vidět pro lineární a nespojitou funkci v obrázcích 2a a 2b.

3.3.2 Lineární rekonstrukce

Další tvar rekonstrukce, kterou můžeme použít, je rekonstrukce lineární. Předpokládáme, že interpolační funkce má tvar

$$f_i(x) = f_i + S_i \cdot (x - x_i) \quad (9)$$

kde jako S_i označujeme směrnici lineární funkce, kterou jsme touto rekonstrukcí získali. Dosadíme (9) do (6), abychom ověřili konzervativitu.



(a) Konstantní rekonstrukce pro lineární funkci. Vidíme, že rekonstrukce není už při počátečních podmínkách přesná
(b) Konstantní rekonstrukce pro nespojitou funkci.

Obrázek 2: Konstantní rekonstrukce

$$\begin{aligned}
& \frac{1}{V_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f_i + S_i \cdot (x - x_i) \, dx = \\
& = \frac{f_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})}{V_i} + \frac{S_i(x_{i+\frac{1}{2}}^2 - x_{i-\frac{1}{2}}^2)}{2V_i} - \frac{S_i x_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})}{V_i} = \\
& = f_i + \frac{S_i(x_{i+\frac{1}{2}}^2 - x_{i-\frac{1}{2}}^2)}{2(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})} - S_i x_i = \\
& = f_i + \frac{S_i(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}} - 2x_i)}{2} = f_i
\end{aligned}$$

Při odvozování jsme využili definicí (1) a (2). Tímto jsme ukázali, že každá lineární rekonstrukce je konzervativní.

Existuje řada metod pro výpočet směrnic S_i optimalizující průběh funkce například z hlediska přesnosti nebo spojitosti. Zde postupujeme stejně jako v [14] za pomoci nejmenších čtverců. Definujeme proto následující funkcionál popisující odchylku rekonstruované funkce od hodnot v sousedních buňkách.

$$\mathcal{F}(S_i) = \sum_{j \in \{i-1, i+1\}} \left(\frac{\int_{G_j} f_i(x) \, dx}{V_j} - f_j \right)^2 \quad (10)$$

Pro získání směrnic S_i funkcionál nyní minimalizujeme.

$$\begin{aligned}
0 &= \frac{d\mathcal{F}}{dS_i} \\
0 &= 2 \left(\frac{\int_{G_{i-1}} f_i + S_i(x - x_i) dx}{V_{i-1}} - f_{i-1} \right) \left(\frac{\int_{G_{i-1}} x - x_i dx}{V_{i-1}} \right) + \\
&\quad + 2 \left(\frac{\int_{G_{i+1}} f_i + S_i(x - x_i) dx}{V_{i+1}} - f_{i+1} \right) \left(\frac{\int_{G_{i+1}} x - x_i dx}{V_{i+1}} \right) \\
0 &= \left(\frac{\int_{G_{i-1}} f_i + S_i(x - x_i) dx}{V_{i-1}} - f_{i-1} \right) (x_{i-1} - x_i) + \\
&\quad + \left(\frac{\int_{G_{i+1}} f_i + S_i(x - x_i) dx}{V_{i+1}} - f_{i+1} \right) (x_{i+1} - x_i) \\
0 &= (f_i + S_i(x_{i-1} - x_i) - f_{i-1}) (x_{i-1} - x_i) + \\
&\quad + (f_i + S_i(x_{i+1} - x_i) - f_{i+1}) (x_{i+1} - x_i)
\end{aligned}$$

Ve výpočtu jsme využili následující vztah

$$\frac{\int_{G_j} x - x_i dx}{V_j} = \frac{x_{j+\frac{1}{2}}^2 - x_{j-\frac{1}{2}}^2}{2(x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}})} - x_i = \frac{x_{j+\frac{1}{2}} + x_{j-\frac{1}{2}}}{2} - x_i = x_j - x_i \quad (11)$$

Z minimalizace získáme

$$S_i = \frac{(f_i - f_{i-1})(x_{i-1} - x_i) + (f_i - f_{i+1})(x_{i+1} - x_i)}{(x_{i-1} - x_i)^2 + (x_{i+1} - x_i)^2}$$

V případě ekvidistantní sítě vztah vede na [15]:

$$S_i = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}, \quad (12)$$

což není nic jiného než centrální diference v bodě x_i .

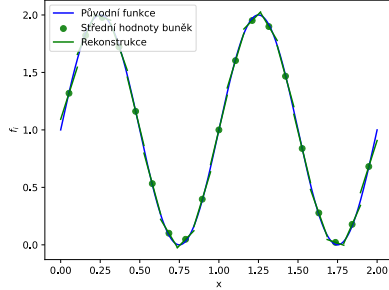
Jak tato rekonstrukce vypadá pro sinus můžeme vidět na obrázku 3a. Pokud tuto rekonstrukci použijeme tak může docházet k problému, kdy nám vznikají extrém, které v původní funkci nebyly. Pro ilustraci tohoto problému se podívejme na obrázek 3b, který nám ukazuje rekonstrukci u nespojitě funkce. Vidíme, že v okolí nespojitosti ve funkci vzniká nový extrém.

3.3.3 Lineární rekonstrukce s limiterem

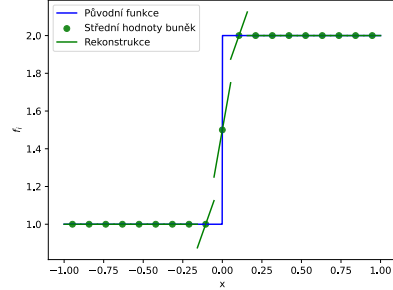
V této podsekcí označme lokální extrém veličiny jako

$$f_i^{max} = \max(f_{i-1}, f_i, f_{i+1}), f_i^{min} = \min(f_{i-1}, f_i, f_{i+1}).$$

Pokud by naše rekonstrukce $f_i(x)$ nabývala větší hodnoty než f_i^{max} , nebo menší hodnoty než f_i^{min} , vznikala by nový extrém, který se projeví oscilacemi



(a) Lineární rekonstrukce sinu



(b) Vznikání nových extrémů - jak vidíme, pokud lineární rekonstrukci použijeme ne daleko rozhraní, vznikají nové extrémy

Obrázek 3: Lineární rekonstrukce

v řešení. Možnost, jak se výše zmíněnému problému vyhnout, je za pomoci takzvaných limiterů [16]. V metodách pro remapování se často využívají Barth a Jaspersenův limiter [17] a Venkatakrishnanův limiter [18]. Zkráceně tyto limity fungují tím způsobem, že zmenší S_i z (12) tak, aby se zabránilo vzniku nových extrémů. Jak jsme zmínili v předchozí sekci tak je každá lineární rekonstrukce konzervativní, proto můžeme S_i změnit.

Fungování Barth a Jaspersenova limiteru (Dále již jen jako BJ limiteru) si ukažme na následujícím příkladě.

Máme buňky $i - 1, i, i + 1$, platí

$$f_{i-1} < f_i = f_{i+1}, x_{i-1} = -\frac{1}{2}, x_i = 0, x_{i+1} = \frac{1}{2}.$$

Platí tedy, že $f_i^{min} = f_{i-1}$ a $f_i^{max} = f_i$ Z (12) máme že

$$S_i = f_{i+1} - f_{i-1} = f_i - f_{i-1} > 0.$$

Vidíme, že pro rekonstrukci vzniká nový extrém (tj. $f_i(x_{i+\frac{1}{2}}) > f_i^{max}$), protože $f_i(x_{i+\frac{1}{2}}) = f_i + S_i \cdot (x_{i+\frac{1}{2}} - x_i) > f_i$. Proto použijeme BJ limiter.

Spočítáme za pomoci původní rekonstrukce následující 2 hodnoty

$$d_l = f_i(x_{i-1}) - f_i, d_r = f_i(x_{i+1}) - f_i$$

BJ limiter je na okrajích buňky pak definován jako

$$\phi_l = \begin{cases} \frac{f_i^{max} - f_i}{d_l} & d_l > 0 \\ \frac{f_i^{min} - f_i}{d_l} & d_l < 0 \end{cases}, \phi_r = \begin{cases} \frac{f_i^{max} - f_i}{d_r} & d_r > 0 \\ \frac{f_i^{min} - f_i}{d_r} & d_r < 0 \end{cases} \quad (13)$$

Z nich můžeme spočítat

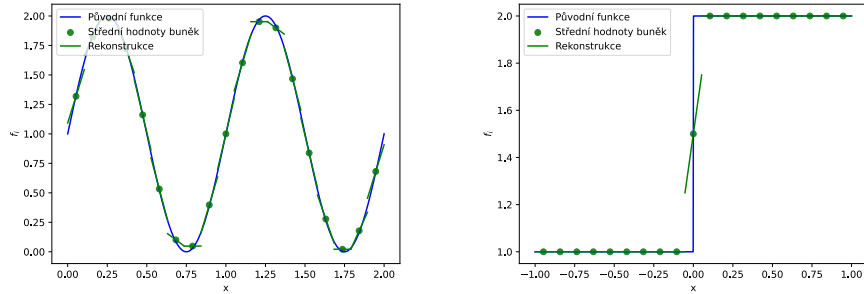
$$\phi_r^{lim} = \min(1, \phi_r), \phi_l^{lim} = \min(1, \phi_l) \quad (14)$$

Pro novou směrnicí pak platí $\hat{S}_i = S_i \cdot \min(1, \phi_l^{lim}, \phi_r^{lim})$. Lze ukázat že tato směrnicí novy extrém nevytváří.

Na obrázku 4b vidíme jak rekonstrukce vypadá. Všimněme si, že nevznikají žádné nové extrém v porovnání s původní nelimitovanou rekonstrukcí. Rozdíl mezi BJ limiterem a Venkatakrišnanovým limiterem je následující. V předposledním kroku BJ limiteru (14) používáme funkci $\min(1, x)$, u Venkatakrišnanova limiteru aproximujeme tuto funkci následující hladkou funkcí

$$\Phi(x) = \frac{x^2 + 2x}{x^2 + x + 2} \quad (15)$$

Venkatakrišnanův limiter se používá v případech, kdy záleží na hladkosti rekonstruované funkce, my ho používáme v Lagrangeovském kroku. Na obrázku 4a vidíme rekonstrukci funkce sinus za pomoci lineární rekonstrukce s BJ limiterem. Všimněme si, že rekonstruovaná funkce má menší extrém než funkce původní. V obrázku 4b vidíme, že díky limiteru nevznikají u rekonstrukce nespojitě funkce nové extrém.



(a) Lineární rekonstrukce funkce sin s BJ limiterem (b) Lineární rekonstrukce skokové funkce s BJ limiterem

Obrázek 4

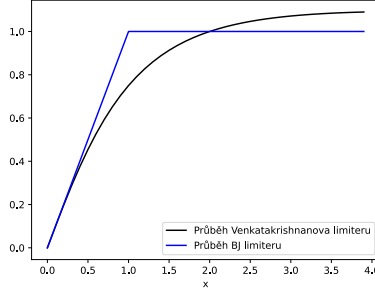
3.3.4 Polynomiální rekonstrukce

Pro remap vyššího stupně lze použít polynomiální rekonstrukci [14]. Pro polynomiální rekonstrukci řádu n předpokládáme její tvar v buňce i jako

$$f_i(x) = f_i + \sum_{j=1}^n S_{i,j} \left((x - x_i)^j - \frac{1}{V_i} \left(\int_{G_i} (x - x_i)^j dx \right) \right), \quad (16)$$

kde $S_{i,j}$ označuje j -tou derivaci v buňce i . Dále označme pravou část závorky jako:

$$K_{i,j} = \frac{1}{V_i} \left(\int_{G_i} (x - x_i)^j dx \right) \quad (17)$$



(a) Porovnání BJ a Venkatakrishnanova limiteru

Obrázek 5

Tato konstanta zaručuje splnění podmínky konzervativity (6). Pokud totiž do podmínky dosadíme, člen v závorce bude roven 0,

$$\begin{aligned}
 & \frac{\int_{G_i} f_i + \sum_{j=1}^n S_{i,j} \left((x - x_i)^j - \frac{1}{V_i} \left(\int_{G_i} (x - x_i)^j dx \right) \right)}{V_i} = \\
 & = f_i + \sum_{j=1}^n S_{i,j} \left(\int_{G_i} (x - x_i)^j dx - \frac{V_i}{V_i} \int_{G_i} (x - x_i)^j dx \right) = \\
 & = f_i
 \end{aligned} \tag{18}$$

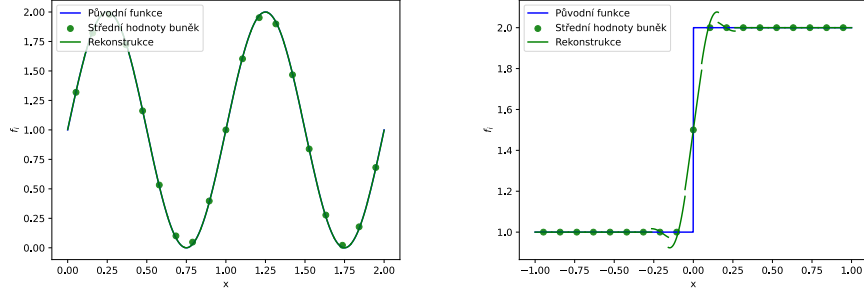
Pro určení koeficientů $S_{i,j}$ existuje několik možností [19, 20, 21]. My postupujeme podle [21] a využijeme metodu nejmenších čtverců. Definujeme funkcionál

$$\mathcal{F}(S_{i,1}, S_{i,2}, S_{i,3}, \dots) = \sum_{k \in \{i - \lceil \frac{n}{2} \rceil, \dots, i + \lceil \frac{n}{2} \rceil\}} \left(\frac{\int_{G_j} f_i(x) dx}{V_j} - f_k \right)^2 \tag{19}$$

kde $\lceil x \rceil$ označuje horní celou část. Tento funkcionál je minimalizován například v [19] za pomoci takzvané Householderovy transformace. Pro případ $n = 2$ dostaneme stejný funkcionál jako jsme dostali pro lineární rekonstrukci (10). V případě polynomů nižšího stupně než n je rekonstruovaná funkce přesná. Díky tomuto je tato metoda vhodná pro rekonstrukci hladkých funkcí. Příklad můžeme vidět na obrázku 6a. Pro funkce obsahující nespojitost vznikají oscilace, jak je vidět na obrázku 6b.

3.3.5 THINC rekonstrukce

Jak jsme tedy zmínili, tak předchozí rekonstrukce nejsou vhodné pro rekonstrukci nespojitých funkcí. K tomuto dochází například na rozhraní dvou ma-



(a) Polynomiální rekonstrukce dobře aproximuje hladké funkce (b) Polynomiální rekonstrukce u nespojitostí vytváří oscilace

Obrázek 6: Polynomiální rekonstrukce

teriálů.

Jeden ze způsobů, jak rekonstrukci na nespojitosti provádět, je za pomoci THINC (Tangent Hyperbola for INTERFACE Capturing method) rekonstrukce [13, 12, 10]. Tato rekonstrukce nám umožňuje popsat plynulý přechod funkce přes nespojitost. THINC rekonstrukci v buňce i najdeme pomocí následujícího vzorce:

$$f_i(x) = \bar{f}_i^{min} + \frac{\bar{f}_i^{max} - \bar{f}_i^{min}}{2} \left[1 + \theta \tanh \left(\beta \left(\frac{x - x_{i-\frac{1}{2}}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} - \bar{x}_i \right) \right) \right], \quad (20)$$

kde definujeme:

$$\bar{f}_i^{min} = \min(f_{i+1}, f_{i-1}), \bar{f}_i^{max} = \max(f_{i+1}, f_{i-1}), \theta = \text{sign}(f_{i+1} - f_{i-1})$$

β je námi zvolená konstanta definující strmou hyperbolického tangensu, pro ilustraci obrázek 7, kde je rekonstrukce zobrazena pro několik různých hodnot β . Poslední neznámá, která ve vzorci zbývá, je \bar{x}_i . Její hodnota je určena [12] díky podmínce (6) jako:

$$\bar{x}_i = \frac{1}{2\beta} \ln \left(\frac{\exp \left(\frac{\beta}{\theta} (1 + \theta - 2\bar{f}_i) \right)}{1 - \exp \left(\frac{\beta}{\theta} (1 - \theta - 2\bar{f}_i) \right)} \right), \bar{f}_i = \frac{f_i - \bar{f}_i^{min}}{\bar{f}_i^{max} - \bar{f}_i^{min}} \quad (21)$$

Tato hodnota posouvá hyperbolický tangens v buňce tak, aby byla zachována konzervativita. Geometrická interpretace [10] tohoto parametru je, že udává inflexní bod rekonstrukce v buňce, pokud ho tedy přešklujeme vztahem $x_{inflexion} = \bar{x}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) - x_{i-\frac{1}{2}}$.

Takto definovaná rekonstrukce má jednu nevýhodu. Díky tomu, jak jsou \bar{f}_i^{min} a \bar{f}_i^{max} definovány, můžeme na hranicích buňky vytvářet nespojitosti.

Tento nedostatek můžeme opravit tím že v okolních buňkách provedeme lineární rekonstrukci (označme $f_{i+1}(x)$ a $f_{i-1}(x)$) a následně předefinujeme.

$$\bar{f}_i^{min} = \min(f_{i+1}(x_{i+\frac{1}{2}}), f_{i-1}(x_{i-\frac{1}{2}})), \bar{f}_i^{max} = \max(f_{i+1}(x_{i+\frac{1}{2}}), f_{i-1}(x_{i-\frac{1}{2}}))$$

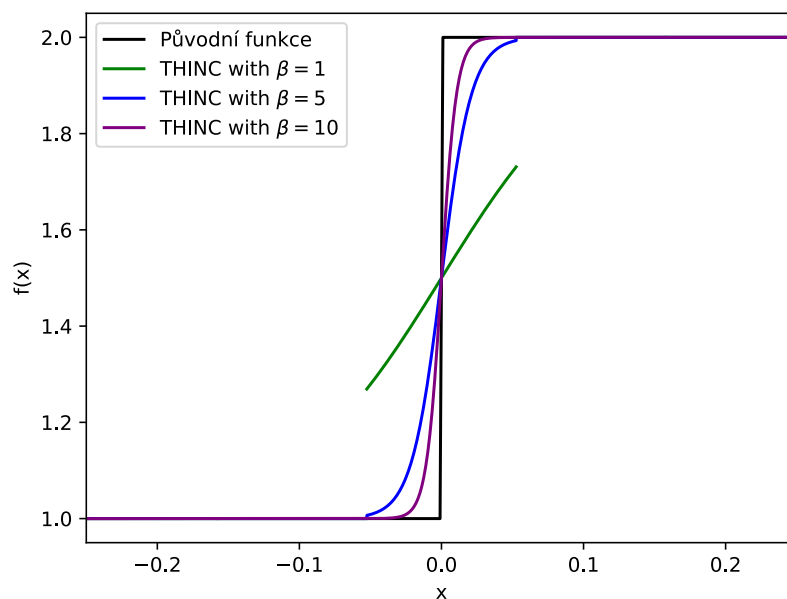
Dále je tato rekonstrukce použitelná pouze v buňkách, kde je funkce lokálně ostře rostoucí či klesající. Tedy

$$f_{i-1} < f_i < f_{i+1} \quad (22)$$

nebo

$$f_{i-1} > f_i > f_{i+1} \quad (23)$$

Z těchto důvodů tuto rekonstrukci nelze použít na celé síťce, ale pouze v buňkách obsahujících nespojitost. Jak tyto buňky určujeme zmiňujeme dále.



Obrázek 7: Porovnání THINC rekonstrukce pro různá β , všimněte si, že pro vysoké β tato rekonstrukce dobře kopíruje průběh nespojité funkce

3.3.6 Rekonstrukce nespojitou funkcí

Jak tedy můžeme na obrázku 7 vidět, tak pro vysoké β rekonstrukce popisuje nespojitost. Zjistíme jak tato rekonstrukce vypadá pro velmi velký parametr β . Chtěli bychom tedy vzít vztah definující THINC rekonstrukci (20) a provést limitu $\beta \rightarrow \infty$. Tedy

$$\lim_{\beta \rightarrow \infty} \bar{f}_i^{min} + \frac{\bar{f}_i^{max} - \bar{f}_i^{min}}{2} \left[1 + \theta \tanh \left(\beta \left(\frac{x - x_{i-\frac{1}{2}}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} - \bar{x}_i \right) \right) \right] \quad (24)$$

Protože pro velké β se funkce tanh z THINC rekonstrukce limitně blíží k znaménkové funkci sign, předpokládáme tvar rekonstrukce jako

$$f_i(x) = \bar{f}_i^{min} + \bar{f}_i^{dif} \Theta(\theta(x - \hat{x})), \quad (25)$$

kde stejně jako u THINC rekonstrukce definujeme

$$\bar{f}_i^{min} = \min(f_{i+1}, f_{i-1}), \bar{f}_i^{max} = \max(f_{i+1}, f_{i-1}), \theta = \text{sign}(f_{i+1} - f_{i-1})$$

a \bar{f}_i^{dif} definujeme jako

$$\bar{f}_i^{dif} = \bar{f}_i^{max} - \bar{f}_i^{min}$$

Navíc označíme jako Θ Heavysideovu funkci:

$$\Theta(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \quad (26)$$

Rekonstrukce tedy závisí pouze na parametru \hat{x} , který určuje místo skoku.

Abychom ukázali spojitost mezi (20) a (25) zapišme si ještě jednu THINC rekonstrukci (označme ji jako f_{sign}), ale místo funkce tanh použijeme znaménkovou funkci. Pro přehlednost označme argument tanh jako $C(x)$

$$f_{sign}(x) = \bar{f}_i^{min} + \frac{\bar{f}_i^{max} - \bar{f}_i^{min}}{2} [1 + \theta \text{sign}(C(x, \theta))] \quad (27)$$

Využíváme následující vztah mezi znaménkovou a Heavysideovou funkcí

$$\Theta(x) = \frac{1 + \text{sign}(x)}{2} \quad (28)$$

a dostaneme

$$f_{sign}(x) = \bar{f}_i^{min} + \bar{f}_i^{dif} \theta \Theta(C(x, \theta)) \quad (29)$$

Což pro $\theta = 1$ je stejné jako (25). Parametr \hat{x} určíme z podmínky konzervativity (6) dosazením

$$f_i = \frac{\int_{G_i} f_i(x) dx}{V_i} = \frac{\int_{x_{i-1}}^{x_{i+1}} \bar{f}_i^{min} + \bar{f}_i^{dif} \Theta(\theta(x - \hat{x})) dx}{x_{i+1} - x_{i-1}} \quad (30)$$

$$f_i = \bar{f}_i^{min} + \frac{\int_{x_{i-1}}^{x_{i+1}} \bar{f}_i^{dif} \Theta(\theta(x - \hat{x})) dx}{x_{i+1} - x_{i-1}} \quad (31)$$

$$f_i - \bar{f}_i^{min} = \frac{\int_{x_{i-1}}^{\hat{x}} \bar{f}_i^{dif} \Theta(\theta(x - \hat{x})) dx}{x_{i+1} - x_{i-1}} + \frac{\int_{\hat{x}}^{x_{i+1}} \bar{f}_i^{dif} \Theta(\theta(x - \hat{x})) dx}{x_{i+1} - x_{i-1}} \quad (32)$$

Vidíme, že θ určuje zda první nebo druhý člen na pravé straně je nula. V případě, kdy $\theta = 1$, platí

$$f_i - \bar{f}_i^{min} = \frac{\int_{\hat{x}}^{x_{i+1}} \bar{f}_i^{dif} dx}{x_{i+1} - x_{i-1}} \quad (33)$$

$$f_i - \bar{f}_i^{min} = \frac{\bar{f}_i^{dif}(x_{i+1} - \hat{x})}{x_{i+1} - x_{i-1}}, \quad (34)$$

z čehož dostaneme

$$\hat{x} = -\frac{V_i(f_i - \bar{f}_i^{min})}{\bar{f}_i^{dif}} + x_{i+1}. \quad (35)$$

Pro případ $\theta = -1$ bychom dostali

$$f_i - \bar{f}_i^{min} = \frac{\int_{x_{i-1}}^{\hat{x}} \bar{f}_i^{dif} \Theta(x - \hat{x}) dx}{x_{i+1} - x_{i-1}}, \quad (36)$$

což nás vede k výsledku

$$\hat{x} = \frac{V_i(f_i - \bar{f}_i^{min})}{\bar{f}_i^{dif}} + x_{i-1}. \quad (37)$$

Spojením obou případů máme jako výsledek:

$$\hat{x} = -\theta \frac{V_i(f_i - \bar{f}_i^{min})}{\bar{f}_i^{dif}} + \Theta(\theta)x_{i+1} + \Theta(-\theta)x_{i-1} \quad (38)$$

Tato rekonstrukce se tedy chová podobně jako THINC rekonstrukce a tak jak jsme ji zavedli má i podobné problémy. Opět na hranicích buněk vytváříme nespojitost, uvažujeme-li pouze střední hodnotu sousedních buněk. Tomuto se ale lze vyhnout tím, že použijeme hodnotu z lineární rekonstrukce v okolních buňkách, tj:

$$\begin{aligned} \bar{f}_i^{min} &= \min(\hat{f}_{i+1}(x_{i+\frac{1}{2}}), \hat{f}_{i-1}(x_{i-\frac{1}{2}})) \\ \bar{f}_i^{max} &= \max(\hat{f}_{i+1}(x_{i+\frac{1}{2}}), \hat{f}_{i-1}(x_{i-\frac{1}{2}})) \\ \bar{f}_i^{dif} &= f_i^{max} - f_i^{min} \end{aligned}$$

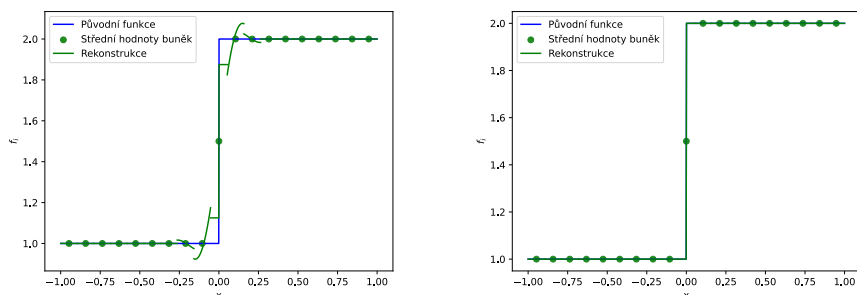
Dále opět tuto rekonstrukci můžeme použít pouze v případě, že jsou střední hodnoty rekonstruované funkce v okolí nespojitosti ryze monotónní.

3.3.7 Rekonstrukce v okolí nespojitosti

Protože obě předchozí rekonstrukce používáme pouze v buňkách, ve kterých se nachází nespojitost, je třeba popsat, jakou rekonstrukci provádíme v okolních buňkách.

Nechceme, aby naše rekonstrukce byla ovlivněna hodnotami v buňkách za nespojitostí. Pro ilustraci proč uvedme následující příklad, viz obrázek 8. Jak vidíme, máme dvě konstantní funkce s rozhraním, které remapujeme. Pokud v buňce vedle rozhraní budeme provádět lineární rekonstrukci, pak podle vzorce (12) bude tato rekonstrukce ovlivněna střední hodnotou buňky, ve které se nachází rozhraní a vzniká nový extrém. Tomuto ale můžeme zamezit pokud použijeme limiter.

Toto samé platí pokud bychom prováděli polynomiální rekonstrukci vyššího stupně v buňkách, které by byly dostatečně blízko nespojitosti.



Případ 1: V buňce vedle rozhraní provádíme polynomiální rekonstrukci příliš velkého řádu. Rekonstrukce je proto ovlivněna hodnotami buněk za nespojitostí a dostí dochází k oscilacím
 Případ 2: Polynomiální rekonstrukci blízko skoku omezujeme tak, aby nebyla ovlivněna buňkami za nespojitostí a tostí

Obrázek 8: Rekonstrukce u nespojitosti

Prakticky tedy v buňce vedle THINC rekonstrukce provádíme lineární rekonstrukci s limiterem, vedle buňky, ve které je lineární rekonstrukce s limiterem, provádíme nejvýše rekonstrukci kvadratickou (to je polynomiální rekonstrukce 2. řádu), vedle kvadratické provádíme polynomiální rekonstrukci nejvýše 4. řádu a tak dále.

Další možností, jak se tomuto problému vyhnout je to, že pro provedení rekonstrukce nepoužijeme symetrické okolí buňky, ale vybíráme buňky tak, aby rekonstrukce přes rozhraní nezasáhla [13]. To znamená, že ve funkcionalu (19) provádíme sumu přes ty buňky, které se všechny nachází na jedné straně rozhraní. V našem kódu jsme však tuto metodu neimplementovali a používáme tedy jen postupně se zvyšující stupeň rekonstrukce popsany výše.

3.3.8 Okrajové podmínky

U každé z předchozích rekonstrukcí jsme předpokládali, že v okolí buňky, kde rekonstrukci provádíme, vždy existuje dostatek buněk. Toto ale samozřejmě neplatí na hranicích výpočetní oblasti. Existuje několik způsobů jak tento problém řešit.

My budeme používat takzvané "ghost" buňky. Toto spočívá v rozšíření výpočetní oblasti o několik buněk, kde jako remapované veličiny zvolíme veličiny buňky z výpočetní oblasti která se k této ghost buňce nachází nejbližší. Další možností je používat jednostranné rekonstrukce, jak je zmíněno například v [20] nebo [13].

3.4 Výpočet hodnot na nové síťce

Jak jsme zmínili v popisu algoritmu pro remap, nakonec potřebujeme získat hodnoty na nové síťce za pomoci integrace námi rekonstruované funkce. Existují 2 ekvivalentní formulace jak toto dělat. V této sekci budeme hodnoty z nové sítě označovat pomocí $\hat{\cdot}$, tj. například objem nové buňky bude označen \hat{V}_i

Jednou z metod je metoda založená na průnicích. V ní integrujeme námi nalezenou globální rekonstrukci $f(x)$ přes jednotlivé objemy nových buněk. Například tedy pro novou střední hodnotu \hat{f}_i buňky i platí

$$\hat{f}_i = \frac{\int_{\hat{G}_i} f(x) dx}{\hat{V}_i} \quad (39)$$

Tento integrál lze vyčíslit jako sumu přes průniky starých buněk s buňkou novou, tj.

$$\hat{f}_i = \frac{\sum_j \int_{G_j \cap G_i} f_j(x) dx}{\hat{V}_i}. \quad (40)$$

V našem kódu využíváme numerickou integraci, u které můžeme nastavit integrační body a nemusíme tedy průniky explicitně konstruovat. Toto ještě můžeme zformulovat v takzvaném tokovém tvaru.

Označme si následující 2 množiny

$$\hat{G}_i^+ = \hat{G}_i \cap G \setminus G_i \quad (41)$$

a

$$\hat{G}_i^- = G_i \cap G \setminus \hat{G}_i \quad (42)$$

\hat{G}_i^+ tedy označuje oblast kterou buňka vyhlazením sítě získala, \hat{G}_i^- označuje oblast kterou naopak ztratila. Nová střední hodnota má pak tvar

$$\hat{f}_i = \frac{1}{\hat{V}_i} \left(f_i V_i + \int_{\hat{G}_i^+} f(x) dx - \int_{\hat{G}_i^-} f(x) dx \right) \quad (43)$$

3.5 Remap v ALE kódu

V ALE simulaci je třeba remapovat všechny veličiny. Remap je ale formulován pouze pro veličiny, které se zachovávají. Je tedy potřeba z primitivních veličin (hustota ρ , tlak p , měrná vnitřní energie ε , rychlost buňky u) sestrojít hustoty a ty remapovat.

V buňce i máme

- Hustotu ρ_i
- Hustotu hybnosti $\mu_i = \rho_i u_i$
- Hustotu kinetické energie $k_i = \rho_i \varepsilon_i + \frac{1}{2} \rho_i u_i^2$

S těmito veličinami provádíme remap. Poté co provedeme integraci na nové síťce veličiny přejdou od hustot k těmto veličinám

- Hustota $\rho_i \mapsto$ celková hmotnost \hat{m}_i
- Hustota hybnosti $\mu_i \mapsto$ hybnost $\hat{m}_i \hat{u}_i$
- Hustota kinetické energie $k_i \mapsto$ celková kinetická energie \hat{E}_{k_i}

Abychom se dostali k novým primitivním veličinám, které používáme pro Lagrangeovský krok, provedeme následující.

- $\hat{\rho}_i = \hat{m}_i / \hat{V}_i$
- $\hat{u}_i = \hat{m}_i \hat{u}_i / \hat{m}_i$
- $\hat{\varepsilon}_i = \frac{\hat{E}_{k_i}}{\hat{m}_i} - \frac{1}{2} \hat{u}_i^2$

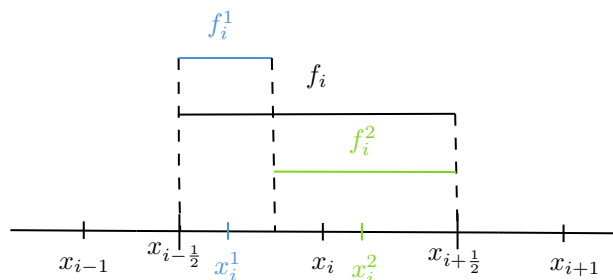
Nakonec spočítáme tlak za pomoci stavové rovnice. Pro námi používanou stavovou rovnici tedy

$$\hat{p}_i = \hat{\rho}_i \hat{\varepsilon}_i (\gamma - 1) \quad (44)$$

3.6 Multimateriálový remap

V případě multimateriálových ALE (MMALE) simulací je třeba používat multimateriálový remap. V multimateriálových ALE simulacích mohou buňky obsahovat více než jeden materiál, každý materiál má svoje veličiny. Pokud máme v simulaci několik velmi rozdílných materiálů, tyto materiály mohou mít rozdílnou stavovou rovnici (například při interakci pevné látky a plynu). Abychom toto mohli modelovat, potřebujeme v buňkách znát rozložení jednotlivých materiálů.

V buňce i tedy oproti jednomateriálovému případu známe navíc střední hodnotu veličin pro dané materiály f_i^m , objemy jednotlivých materiálů V_i^m a středy jednotlivých materiálů x_i^m . Horním indexem m označujeme o jaký materiál se jedná. Navíc předpokládáme, že v buňce může být nejvýše jedno rozhraní mezi dvěma materiály. Pro ilustraci uvádíme obrázek 9, kde je zobrazeno rozdělení buňky mezi 2 materiály, co mají svoji střední hodnotou f_i^1 a f_i^2 .



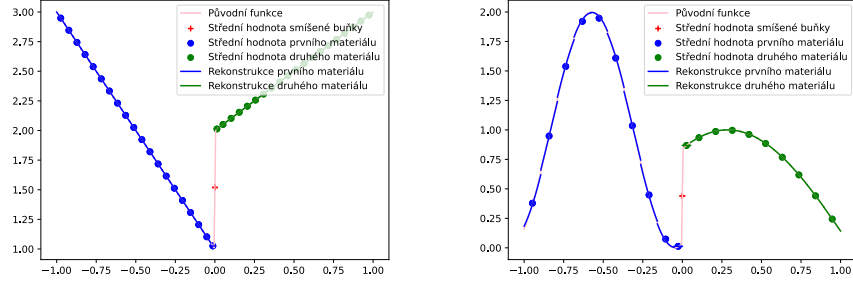
Obrázek 9: Ukázka značení v buňce i u MMALE remapu

Dále zavádíme takzvané *volume fractions* (objemové zlomky), ty jsou v buňce i pro materiál m definované jako

$$\varphi_i^m = \frac{V_i^m}{V_i}, \quad (45)$$

ty určují jak velká část buňky přísluší konkrétním materiálům, na základě jejich objemu. Pro buňky obsahující jeden materiál je tedy objemový zlomek daného materiálu 1 a ostatních materiálů 0. Algoritmus pro multimateriálové ALE metody je stejný jako pro jednomateriálové, tedy opět provádíme Lagrangeovský a Eulerovský krok. V tomto případě ale v remapu, který v Eulerovském kroku provádíme, dochází ke změně. Jak jsme již zmínili v MMALE simulaci známe rozložení materiálů v jednotlivých buňkách, při remapu toto rozložení také musíme remapovat.

Toto můžeme triviálně provést v buňkách, které obsahují pouze jeden materiál. Ve smíšených buňkách musíme provést takzvanou materiálovou rekonstrukci. My na materiálovou rekonstrukci používáme metodu Moment of fluid. [22] V 1D spočívá v seřazení materiálových středů a určení polohy rozhraní na základě objemových zlomků. Při multimateriálové rekonstrukci můžeme použít všechny metody zmíněné u jednomateriálové rekonstrukce, ale THINC a skokovou rekonstrukci nepoužíváme, protože o nespojitosti se postaráme právě díky materiálové rekonstrukci [9]. Pro ukázkou, jak multimateriálová rekonstrukce vypadá, uvádíme obrázky 10a a 10b, u prvního obrázku vidíme rekonstrukci 2 materiálů s lineárním průběhem a rozhraním mezi nimi, u druhého obrázku vidíme rekonstrukci materiálu se sinusoidálním průběhem.



(a) Rekonstrukce 2 materiálů s lineárním průběhem, vidíme že metoda přesně rekonstruuje tento problém

(b) Rekonstrukce 2 materiálů se sinusoidálním průběhem, ve všech buňkách provádíme polynomiální rekonstrukci 4. řádu

Obrázek 10: Multimateriálové rekonstrukce

4 Lagrangeovský krok ALE metod

Pro Lagrangeovský krok jsme zvolili cell-centered metodu, toto nám umožňuje remapovat všechny proměnné stejně.

4.1 Fyzikální popis

V 1D hydrodynamice popisujeme pohyb stlačitelné tekutiny následujícími Eulerovými rovnicemi.

Předpokládáme fluidní částici.[13] Označme její počáteční pozici jako X a označme její pozici v čase t jako $x(X, t)$. X označuje její Lagrangeovu souřadnici. Známe-li funkci $u = u(x, t)$, udávající rychlost tekutiny, tak pak celou trajektorii můžeme spočítat ze vztahu

$$\frac{\partial x}{\partial t} = u, x(X, 0) = X, \quad (46)$$

kde x udává Eulerovu souřadnici.

1D systém Eulerovských rovnic pro hydrodynamiku lze zapsat jako

$$\rho \frac{d}{dt} \frac{1}{\rho} - \frac{\partial}{\partial x} u = 0 \quad (47)$$

$$\rho \frac{d}{dt} u + \frac{\partial}{\partial x} p = 0 \quad (48)$$

$$\rho \frac{d}{dt} E + \frac{\partial}{\partial x} (pu) = 0 \quad (49)$$

kde ρ označuje hustotu, p tlak, u rychlost a E měrnou celkovou energii. Dále symbolem $\frac{d}{dt}$ rozumíme totální (v literatuře také nazývanou jako materiálovou) derivaci.

Pro uzavření systému potřebujeme ještě jednu rovnici. K tomu slouží stavová rovnice ideálního plynu.

$$p = P(\rho_i, \varepsilon_i) = \rho\varepsilon(\gamma - 1), \quad (50)$$

kde γ je adiabatická konstanta a ε měrná vnitřní energie $\varepsilon = E - \frac{1}{2}\|u\|^2$.

4.2 Diskretizace prvního řádu

V síti definované v sekci 2.1 je pohyb uzlu $x_{i+\frac{1}{2}}$ popsán podle (46) jako

$$\frac{\partial x_{i+\frac{1}{2}}}{\partial t} = u_{i+\frac{1}{2}}, x_{i+\frac{1}{2}}(0) = X_{i+\frac{1}{2}} \quad (51)$$

Označme střední hodnotu hustoty jako ρ_i , střední hodnotu rychlosti jako u_i a střední hodnotu měrné celkové energie E_i .

Integrujeme systém rovnic (47) (48) (49), to spolu s vlastností Lagrangeovské formulace, kdy je hmota buňky konstantní (tj. $m_i = \rho_i(t)V_i(t), \forall t$), dává [23]:

$$m_i \frac{d}{dt} \frac{1}{\rho_i} + u_{i-\frac{1}{2}} - u_{i+\frac{1}{2}} = 0 \quad (52)$$

$$m_i \frac{d}{dt} u_i - p_{i-\frac{1}{2}} + p_{i+\frac{1}{2}} = 0 \quad (53)$$

$$m_i \frac{d}{dt} E_i - p_{i-\frac{1}{2}} u_{i-\frac{1}{2}} + p_{i+\frac{1}{2}} u_{i+\frac{1}{2}} = 0 \quad (54)$$

kde $u_{i-\frac{1}{2}}$ a $p_{i-\frac{1}{2}}$ označují toky uzly.

Tyto toky můžeme dostat řešením Riemannova problému na rozhraní mezi buňkami. Na levé straně uzlu máme stav $\frac{1}{\rho_i}, u_i, E_i$, na pravé $\frac{1}{\rho_{i+1}}, u_{i+1}, E_{i+1}$. Pro tuto aproximaci používáme Godunovův akustický řešič [24]. Definujeme akustickou impedanci $z_i = \rho_i a_i$, kde a označuje rychlost zvuku. Řešení Riemannova problému je pak dáno následovně:

$$u_{i+\frac{1}{2}} = \frac{z_i u_i + z_{i+1} u_{i+1}}{z_i + z_{i+1}} - \frac{p_{i+1} - p_i}{z_i + z_{i+1}} \quad (55)$$

$$p_{i+\frac{1}{2}} = \frac{z_i p_{i+1} + z_{i+1} p_i}{z_i + z_{i+1}} - \frac{z_i z_{i+1} (u_{i+1} - u_i)}{z_i + z_{i+1}} \quad (56)$$

Provedeme diskretizaci rovnic (52) (53) (54) v čase. Známe všechny proměnné v čase $t = t^n$. Definujeme $\Delta t = t^{n+1} - t^n$.

Poté nahradíme derivace pro každou veličinu dopřednými diferencemi

$$\frac{\partial g(t)}{\partial t} \approx \frac{g^{n+1} - g^n}{\Delta t} \quad (57)$$

Dostáváme

$$m_i \left(\frac{1}{\rho_i^{n+1}} - \frac{1}{\rho_i^n} \right) + \Delta t \left(u_{i-\frac{1}{2}}^n - u_{i+\frac{1}{2}}^n \right) = 0 \quad (58)$$

$$m_i \left(u_i^{n+1} - u_i^n \right) - \Delta t \left(p_{i-\frac{1}{2}}^n + p_{i+\frac{1}{2}}^n \right) = 0 \quad (59)$$

$$m_i \left(E_i^{n+1} - E_i^n \right) - \Delta t \left(p_{i-\frac{1}{2}}^n u_{i-\frac{1}{2}}^n + p_{i+\frac{1}{2}}^n u_{i+\frac{1}{2}}^n \right) = 0 \quad (60)$$

Tato sada diskretních rovnic vede na schéma prvního řádu v čase i prostoru. Pohyb uzlů je pak dán jako

$$x_{i+\frac{1}{2}}^{n+1} = x_{i+\frac{1}{2}}^n + \Delta t u_{i+\frac{1}{2}}^n \quad (61)$$

Dále musíme spočítat novou měrnou vnitřní energii, $\varepsilon_i^{n+1} = E_i^{n+1} - \frac{1}{2}(u_i^{n+1})^2$. Nakonec získáme tlak ze stavové rovnice (50).

4.3 Diskretizace druhého řádu

Vylepšení schématu na 2. řád přesnosti je popsáno v [23].

Předpokládáme, že funkce udávající rychlost v buňce i je lineární, stejně jako u eulerovského kroku provádíme rekonstrukci za pomoci nejmenších čtverců a sousedních buněk.

Opět nechceme, aby vznikali nové extrémy, znovu tedy používáme limiter. Narozdíl od remapu zde používáme Venkatakrishnanův limiter [18], který má hladší průběh než BJ limiter.

Rovnice (58) (59) (60) pak nahradíme rovnicemi středovanými v čase,

$$m_i \left(\frac{1}{\rho_i^{n+1}} - \frac{1}{\rho_i^n} \right) + \Delta t \left(u_{i-\frac{1}{2}}^{n+\frac{1}{2}} - u_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) = 0 \quad (62)$$

$$m_i \left(u_i^{n+1} - u_i^n \right) - \Delta t \left(p_{i-\frac{1}{2}}^{n+\frac{1}{2}} + p_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) = 0 \quad (63)$$

$$m_i \left(E_i^{n+1} - E_i^n \right) - \Delta t \left(p_{i-\frac{1}{2}}^{n+\frac{1}{2}} u_{i-\frac{1}{2}}^{n+\frac{1}{2}} + p_{i+\frac{1}{2}}^{n+\frac{1}{2}} u_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) = 0 \quad (64)$$

Pohyb uzlu je pak popsán

$$x_{i+\frac{1}{2}}^{n+1} = x_{i+\frac{1}{2}}^n + \Delta t u_{i+\frac{1}{2}}^{n+\frac{1}{2}} \quad (65)$$

4.4 Velikost časového kroku

Pro zaručení stability metody volíme velikost časového kroku následovně [23].

Z CFL podmínky máme

$$\Delta t_{CFL} = C_{CFL} \min_i \frac{\Delta x_i^n}{a_i^n}, \quad (66)$$

kde C_{CFL} je námi určený koeficient větší než 0 a a_i^n je rychlost zvuku v buňce i .

Dále použijeme podmínku omezující změnu objemu buňky.

$$\Delta t_V \leq C_V * \frac{V_i^n}{|u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}|}, \quad (67)$$

kde C_V je konstanta z intervalu $(0, 1)$

Dále používáme omezení na změnu časového kroku

$$\Delta t_m = C_M \Delta t^{n-1}, \quad (68)$$

kde C_M je námi určená konstanta větší než 1, díky které se časový krok postupně zvětšuje.

Nový časový krok je pak určen

$$t^n = \min(\Delta t_{CFL}, \Delta t_V, \Delta t_m) \quad (69)$$

V našem ALE kódu konstanty volíme následovně:
 $C_{CFL} = 0.25, C_V = 0.1, C_M = 1.01$.

4.5 Okrajové podmínky

Protože v našem kódu používáme diskretizaci druhého řádu, která provádí rekonstrukci veličin, je potřeba aby všechny buňky vedle sebe měly souseda. Toto neplatí na kraji výpočetní oblasti, proto opět používáme ghost buňky stejně jako u remapu.

V případě, že simulaci zastavíme před tím, než vlny dorazí na okraj simulované oblasti, jsou uzly na hranici udržené v klidu díky rovnováze mezi krajní buňkou a sousedící ghost buňkou.

Pokud to problém vyžaduje můžeme rychlost okrajových uzlů nastavit ručně. Například pro levý krajní uzel dáváme $u_{-\frac{1}{2}} = u_L(t)$, pro funkci $u_L(t) = 0$ se jedná o zeď, pro $u_L(t) = at$ se jedná o rovnoměrně zrychlující píšť.

4.6 Sledování kontaktu

Protože v Eulerovském kroku provádíme remap za pomoci nespojitých funkcí, který můžeme provést pouze na rozhraní, je potřeba označit buňku, ve které se kontaktní nespojitost nachází.

My kontakt sledujeme následovně. Z počátečních podmínek známe pozici kontaktu jako x_c^0 . Předpokládáme, že kontakt se nachází v buňce i . Na konci každého Lagrangeovského kroku pak provádíme následující aktualizaci jeho pozice.

$$x_c^{n+1} = x_c^n + \Delta t \left(\left(1 - \frac{x_c^n - x_{i-\frac{1}{2}}^n}{V_i^n} \right) u_{i-\frac{1}{2}}^{n+\frac{1}{2}} + \frac{x_c^n - x_{i-\frac{1}{2}}^n}{V_i^n} u_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (70)$$

Tedy lineárně interpolujeme rychlost uzlů buňky ve které se kontakt nachází. V našem kódu si pro každý kontakt který sledujeme, ukládáme jeho současnou polohu a u každého Eulerovského kroku explicitně zjišťujeme v které buňce se kontakt nachází. V Eulerovském kroku pak v buňce, která nespojitost obsahuje, provádíme THINC nebo nespojitou rekonstrukci.

4.7 Detekce komprese

Rázová vlna většinou bývá roztažena přes dvě až čtyři buňky, proto není vhodná pro THINC rekonstrukci. V remapu ale u této nespojitosti nechceme provádět rekonstrukci vysokého řádu, proto si buňky, které kompresní vlnu obsahují, potřebujeme označit.

K tomu používáme tento detektor

$$\left(u_{i+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right) < (\kappa - 1) \frac{V_i^n}{\Delta t} \quad (71)$$

κ je námi zvolený parametr, v našem kódu volíme $\kappa = 0.95$.

Pokud je tato nerovnost pravdivá, buňku označíme a u remapu v této buňce používáme lineární rekonstrukci s limiterem.

5 Implementace

Náš ALE kód jsem následně implementoval v programovacím jazyce Python 3.

Python je open source vysokoúrovňový multiparadigmatický programovací jazyk. Jeho první verze byla vydána v roce 1991. První verze Pythonu 3 byla vydána v roce 2008, ale jazyk je stále dále vyvíjen, poslední verze vydána v době psaní je Python 3.11. Python je dynamicky typovaný a garbage collected jazyk. Existuje mnoho implementací tohoto jazyka, my jsme zvolili tu nejpoužívanější, CPython, což je interpreter psaný v jazyce C.

Tento jazyk jsme zvolili z následujících důvodů. Hlavním důvodem bylo to, že s Pythonem jsme měli na začátku bakalářské práce nejvíce zkušeností. Dále Python, protože jeho hlavní implementace je interpretovaná, obsahuje velmi dobrý debugger. Jako poslední důvod uveďme to, že k Pythonu je dostupné velké množství balíků (knihoven) pro vědecké počítání.

Kód samotný jsme psali v objektově orientovaném stylu. Tento styl jsme zvolili, protože jsme chtěli využívat zapouzdřování a abstrakce, kterou tento styl přináší.

5.1 Použité knihovny

Pro výpočet integrálů je možné použít analytické metody, my ale integrujeme numericky. Pro numerickou integraci jsme použili funkci `quad` z balíku `scipy.integrate`. Tato funkce funguje jako wrapper pro funkci `QAGS` z knihovny `QUADPACK` pro jazyk `FORTRAN 77`. Celkově je tato metoda postavena na adaptivní Gaussových kvadraturách [25], tedy pro naše polynomiální rekonstrukce je přesná.

Pro výpočet koeficientů S_i pro polynomiální rekonstrukci využíváme funkci pro výpočet pseudoinverzní matice `pinv` z balíku `scipy.linalg`. Tato metoda je wrapper pro funkci `DGELSS` z knihovny `LAPACK`. Tato metoda vyřeší úlohu nejmenších čtverců za pomoci singulárního rozkladu (SVD).

Pro kreslení obrázků jsme využili balíku `matplotlib`.

6 Numerické výsledky - remap

Naše remapovací metoda je navržena na použití spolu s vyhlazováním sítě v ALE simulacích. Nejdříve ale metodu otestujeme bez Lagrangeovského kroku.

Pro testování přesnosti remapovací metody používáme postup cyklického remapování [3]. V něm zvolíme nějakou funkci, kterou budeme remapovat, tu diskretizujeme, poté námi zvoleným způsobem sítě hýbáme a diskretizované hodnoty remapujeme na posunutou síťku. Toto opakujeme a nakonec porovnáme remapované hodnoty s hodnotami původními.

Parametr, který určuje počet opakování pohybu sítě a následného remapu, zde značíme t_{max} . Pro aktuální časovou vrstvu používáme symbol t_c .

6.1 Hladký pohyb sítě

Při hladkém pohybu sítě [26] je poloha uzlu výpočetní sítě $x_{i-\frac{1}{2}}$ definována vztahem

$$x_{i-\frac{1}{2}}(x) = (1 - \alpha(t))\xi(i - \frac{1}{2}) + \alpha(t)\xi(i - \frac{1}{2})^3, 0 \leq \xi \leq 1, 0 \leq t \leq 1 \quad (72)$$

kde

$$\alpha(t) = \frac{\sin(4\pi t)}{2}, t = \frac{t_c}{t_{max}} \quad (73)$$

a

$$\xi(i - \frac{1}{2}) = \frac{i - 1}{n - 1} \quad (74)$$

Vidíme že pro $\alpha(0)$ a $\alpha(1)$ nám rovnice (72) dává stejnou síťku. Můžeme tedy přímo porovnat rozdíl mezi počátečními a konečnými středními hodnotami v buňkách. Chybu počítáme za pomoci následujících vzorců [3]

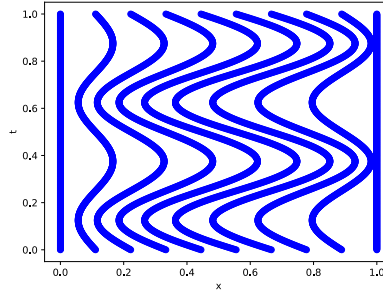
$$L_1 = \frac{\sum_i |f_i - f_{i0}| V_i}{\sum_i |f_i V_i|} \quad (75)$$

$$L_2 = \frac{\sum_i (f_i - f_{i0})^2 V_i}{\sum_i |f_i^2 V_i|}, \quad (76)$$

kde f_{i0} označuje hodnotu v buňce v čase $t = 0$

Průběh tohoto pohybu sítě je vyobrazený na obrázku 11

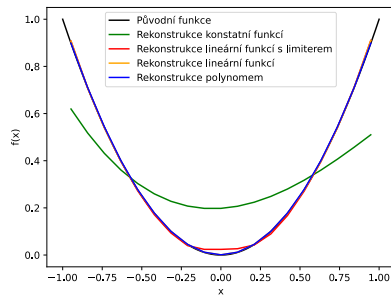
V literatuře [13][3] se nejčastěji používá právě tento cyklický remap. Toto je hlavně z důvodů jednoduché reprodukce výsledků z důvodů explicitně popsání pohybu sítě (72). V této práci budeme používat stejný postup, abychom mohli porovnat naše hodnoty s dříve publikovanými výsledky.



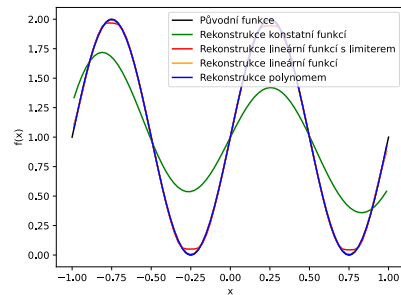
Obrázek 11: Pohyb sítě pro cyklickou rekonstrukci

6.2 Kvadratický polynom a sinus

Ukažme si nejprve výsledek pro cyklický remap s různými rekonstrukcemi pro kvadratickou funkci x^2 (Obrázek 12a) a pro funkci sinus $1 + \sin(2\pi x)$ (Obrázek 12b) převzanou z [3]. Všimněme si toho, jak jsou která schémata numericky difúzní. Nejvíce difúzní je konstantní rekonstrukce. U limiterované rekonstrukce si všimněme, že extrémní funkce se zmenšují. Remap s polynomiální rekonstrukcí je pro kvadratický polynom přesný, pro sinus nastává velmi malá chyba. L1 a L2 chyby pro různé rozlišení sítě při remapování funkce sinus jsou uvedeny v následujících tabulkách 1 a 2.



(a) Remap pro kvadratický polynom



(b) Remap pro sinus

Obrázek 12: Remap spojitéch funkcí, počet buněk je 80 a provedli jsme 300 iterací

L1	40 buněk	80 buněk	160 buněk
Konstantní rekonstrukce	4.7×10^{-1}	3.0×10^{-1}	1.0×10^{-1}
Lineární rekonstrukce	3.5×10^{-2}	4.3×10^{-3}	6.7×10^{-4}
Lineární rekonstrukce s BJ limiterem	5.3×10^{-2}	1.2×10^{-2}	2.3×10^{-3}
Polynomiální rekonstrukce 4. stupně	8.2×10^{-4}	2.3×10^{-5}	6.4×10^{-7}

Tabulka 1: L1 chyby pro remap funkce sinus, provádíme 5krát více remapů než je počet buněk, tj, pro 40 buněk provádíme 200 remapů

L2	40 buněk	80 buněk	160 buněk
Konstantní rekonstrukce	1.9×10^{-1}	7.6×10^{-2}	8.8×10^{-3}
Lineární rekonstrukce	1.1×10^{-3}	3.3×10^{-5}	3.8×10^{-6}
Lineární rekonstrukce s BJ limiterem	3.7×10^{-3}	2.1×10^{-4}	1.1×10^{-5}
Polynomiální rekonstrukce 4. stupně	6.1×10^{-7}	6.0×10^{-10}	1.8×10^{-12}

Tabulka 2: L2 chyby pro remap funkce sinus, provádíme 5krát více remapů, než je počet buněk

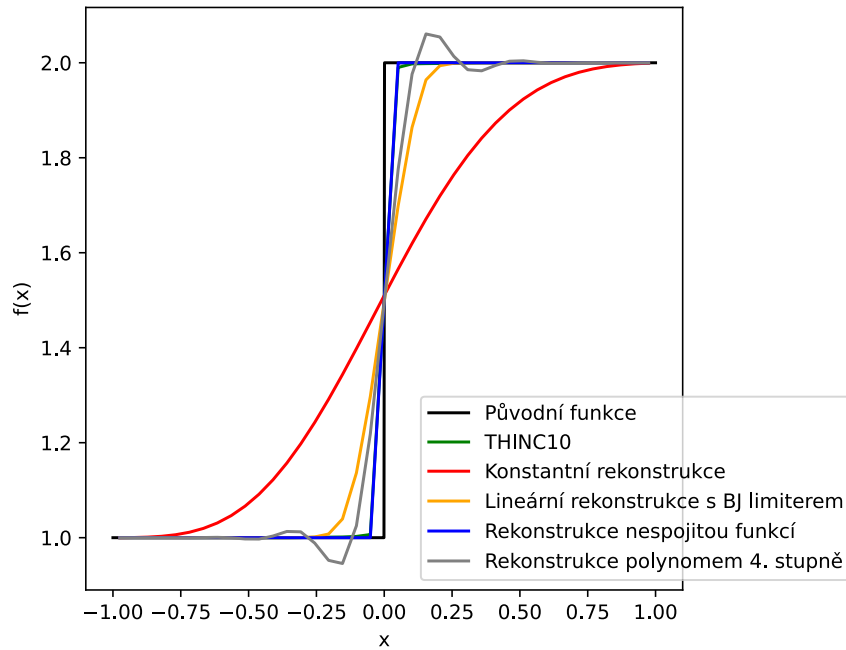
6.3 Nespojitá funkce

Ukažme si dále výsledek cyklického remapu s různými rekonstrukcemi u nespojité funkce

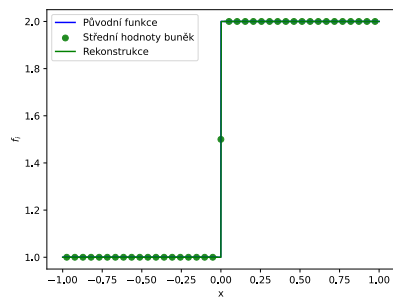
$$g(x) = \begin{cases} 2 & ; x > 0 \\ 1 & ; x \leq 0 \end{cases} \quad (77)$$

. Výsledek vidíme na obrázku 13a, opět vidíme jak jsou které rekonstrukce difúzní. Navíc vidíme nové extrémy, které vznikly pro nelimitovanou lineární rekonstrukci a pro rekonstrukci kvartickým polynomem (polynomem 4. stupně). Toto je hlavní problém při remapování veličin, kterému se snažíme vyhnout. Dále v obrázku vidíme rekonstrukci THINC pro $\beta = 10$.

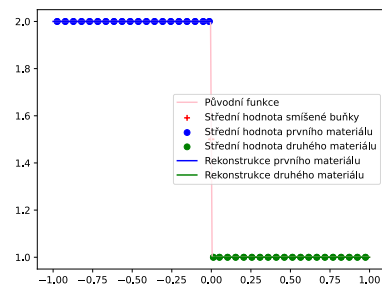
Námi navržená rekonstrukce nespojitou funkcí (Obrázek 13b) a multiteriálová rekonstrukce (Obrázek 13c), v tomto konkrétním případě přesně rekonstruuje původní funkci, chyba pro ně vychází na úrovni počítačové přesnosti.



(a) Různé typy rekonstrukcí



(b) Námí navržená rekonstrukce nespojitou funkcí



(c) Multimateriálová rekonstrukce

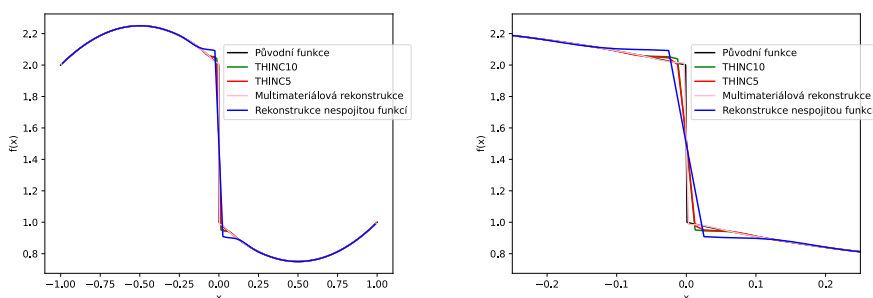
Obrázek 13: Remap nespojitě funkce, počet buněk je 40 a provedli jsme 300 iterací

6.4 Nespojité nelineární funkce

V obrázku 14 vidíme porovnání nepolynomiálních rekonstrukcí a multimateriálové rekonstrukce pro cyklický remap nespojité nelineární funkce. Tato funkce je zadána jako

$$g(x) = \begin{cases} x^2 - x + 1; & x > 0 \\ -x^2 - x + 2; & x \leq 0 \end{cases} \quad (78)$$

V tabulce 3 můžeme vidět L1 a v tabulce 4 L2 chyby pro různý počet buněk. Pokud bychom si z posledních dvou sloupců spočítali řád přesnosti jednotlivých metod, dostali bychom, že metody jsou druhého řádu.



(a) Přiblížení nespojitosti

Obrázek 14: Remap nespojité nelineární funkce, počet buněk je 80 a provedli jsme 300 iterací

L1	40 buněk	80 buněk	160 buněk
Multimateriálová rekonstrukce	1.7×10^{-3}	5.2×10^{-4}	1.3×10^{-4}
Rekonstrukce nespojitou funkcí	7.9×10^{-3}	2.9×10^{-3}	7.2×10^{-4}
THINC $\beta = 10$	8.4×10^{-3}	3.0×10^{-3}	8.8×10^{-4}
THINC $\beta = 5$	7.8×10^{-3}	2.7×10^{-3}	6.5×10^{-4}

Tabulka 3: L1 chyby pro remap nespojité nelineární funkce, provádíme 5krát více remapů, než je počet buněk

L2	40 buněk	80 buněk	160 buněk
Multimateriálové rekonstrukce	9.8×10^{-6}	1.5×10^{-6}	2.0×10^{-7}
Rekonstrukce nespojitou funkcí	3.2×10^{-4}	7.5×10^{-5}	8.9×10^{-6}
THINC $\beta = 10$	3.3×10^{-4}	8.0×10^{-5}	1.1×10^{-5}
THINC $\beta = 5$	2.5×10^{-4}	5.2×10^{-5}	5.9×10^{-6}

Tabulka 4: L2 chyby pro remap nespojité nelineární funkce, provádíme 5krát více remapů, než je počet buněk

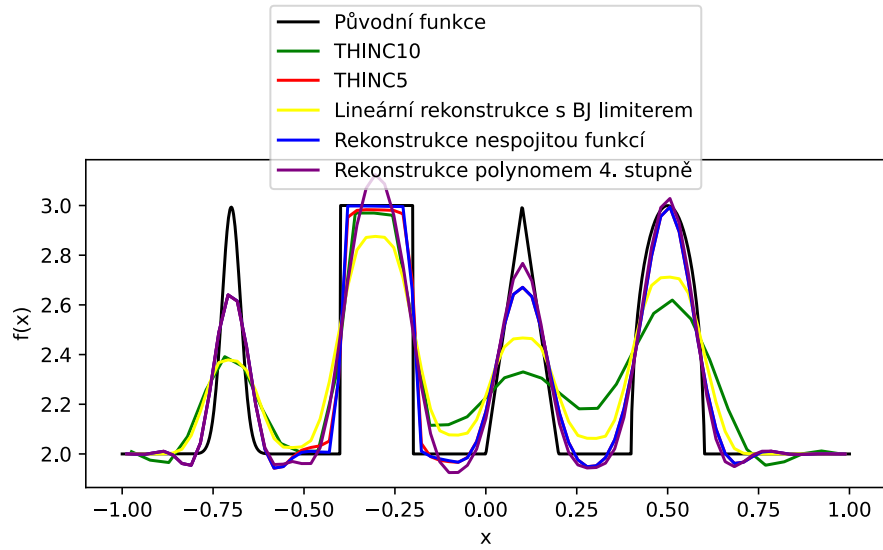
Ukažme si výsledek cyklického remapu pro funkci se čtyřmi profily, přebrané z [13]. Předpis této funkce je

$$h(x) = 1 + \begin{cases} 1 + (G(x, \beta, -0.705) + G(x, \beta, -0.695) + 4G(x, \beta, -0.7)) & x \in \langle -0.8, -0, 6 \rangle \\ 2 & x \in \langle -0.4, -0, 2 \rangle \\ 2 - |10(x - 0.1)| & x \in \langle 0.0, 0, 2 \rangle \\ 1 + (F(x, 10, 0.495) + F(x, 10, 0.505) + 4F(x, 10, 5))/6 & x \in \langle 0.4, 0, 6 \rangle \\ 1 & \text{jinak} \end{cases} \quad (79)$$

kde $\beta = \log(\frac{2}{36 * 0.005^2})$, $G(x, \beta, z) = \exp(-\beta(x - z)^2)$
a $F(x, \alpha, a) = \sqrt{\max(1 - \alpha^2(x - a)^2, 0)}$

Testovali jsme funkce THINC a naši nespojitou funkci, výsledek vidíme na obrázku 15. Tento nepolynomiální remap jsme dělali v buňkách obsahující nespojitost v bodech -0.4 nebo -0.2. Dále abychom zabránili vzniku oscilací, provedli jsme v buňkách obsahující body 0, 0.1, 0.2, 0.4 nebo 0.6 provedli lineární remap s BJ limiterem. V ostatních buňkách děláme polynomiální rekonstrukci 4. řádu. Dále v tom stejném obrázku vidíme výsledky remapu u kterého jsme prováděli buď pouze polynomiální rekonstrukci 4. stupně, lineární rekonstrukci nebo lineární rekonstrukci s limiterem.

Vídíme, že nepolynomiální rekonstrukce v tomto případě dosahují velice dobrých výsledků na rozhraní, u kterého se nachází.



Obrázek 15: Remap funkce s profily, počet buněk je 80 a provedli jsme 300 iterací

7 Numerické výsledky - ALE metody

V této kapitole ukážeme výsledky vybraných hydrodynamických testů získaných v nově implementovaném 1D ALE kódu.

Tyto testy můžeme běžet ve třech režimech:

1. Lagrangeovský režim - neprovádíme remap
2. Eulerovský režim - remapujeme po každém kroku na původní síťku
3. ALE režim - každých 20 kroků síťku Laplaceovsky vyhladíme a remapujeme

Eulerovský režim můžeme používat pouze pokud se okraje naší výpočetní oblasti nehýbou.

Test	Výpočetní oblast	Poloha rozhraní	Stav nalevo od rozhraní			Stav napravo od rozhraní			Koncový čas t_{fin}	Gamma γ
			ρ	u	p	ρ	u	p		
Sod	(0,1)	0.5	1	0	1	0.125	0	0.1	0.2	1.4
Lax	(0,1)	0.5	1	0	1000	1	0	0.01	0.012	1.4
Noh	(-1,1)	0	1	1	10^{-6}	1	-1	10^{-6}	0.6	1.66666
Shu-Osher	(-9.5, 4.5)	-4	3.857143	2.629369	10.333333	$1 + 0.2 \sin(5\pi x)$	0	1	0.18	1.4

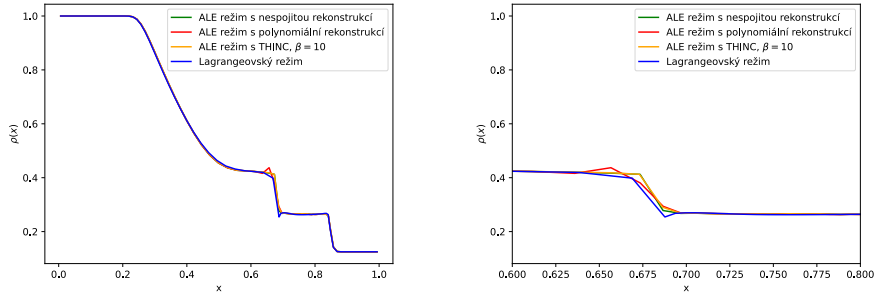
Tabulka 5: Počáteční podmínky pro vybrané testy

7.1 Riemmanovy problémy

Sodův test [13] se řadí mezi Riemmanovy problémy, tedy problémy jejichž počáteční podmínky jsou dané jako po částech konstantní funkce. Počáteční podmínky pro vybrané Riemmanovy testy najdeme v tabulce 5.

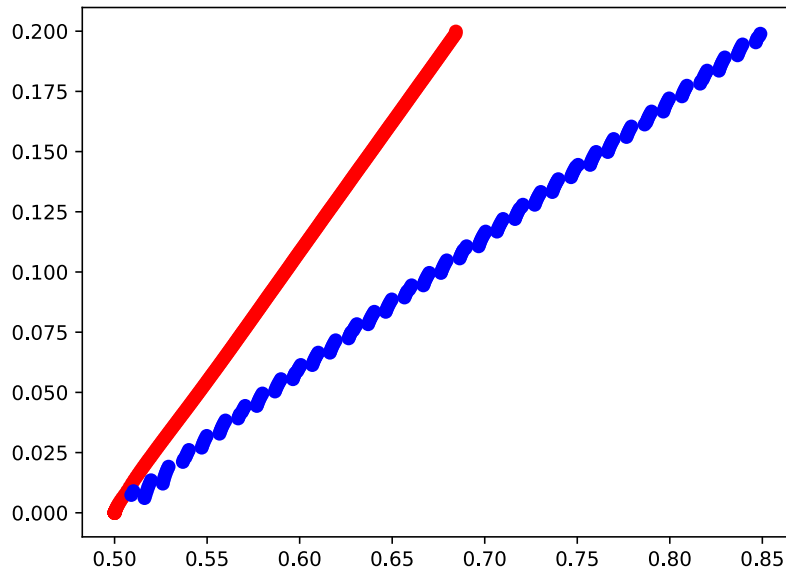
U těchto problémů volíme jako okrajové podmínky zdi, krajní uzly se tedy nehýbou a výpočetní oblast zůstává stejná po celou dobu výpočtu. Na obrázku 16 ukazujeme Sodův problém vyřešený za pomoci ALE metod s různými rekonstrukcemi. Všimněme si, že naše rekonstrukce dává velmi podobné výsledky jako rekonstrukce pomocí THINC.

Na obrázku 17 ukazujeme pohyb sledované nespojitosti u Sodova problému a detekci rázové vlny. Dále na obrázku 18 ukazujeme ALE simulaci Sodova problému pro různé rozlišení výpočetní sítě při použití nespojitě rekonstrukce.

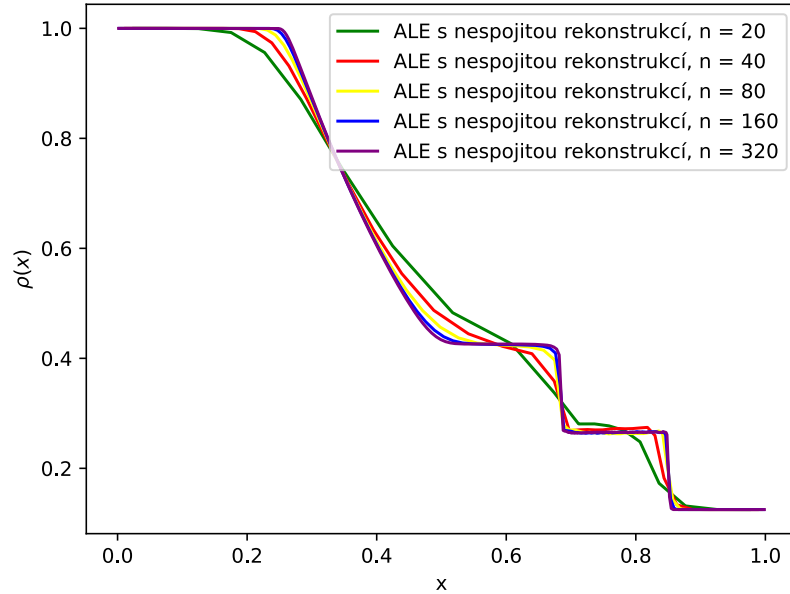


(a) Přiblížení nespojitosti

Obrázek 16: Výsledky numerického řešení Sodova problému, problém jsme řešili různými režimy na síťce s 80 buňkami



Obrázek 17: Zde vidíme jak funguje sledování nespojitosti a detekce komprese u Sodova problému. Červená označuje pohyb sledované nespojitosti, modrá, která se nachází napravo od ní, představuje rázovou vlnu detekovanou detektorem komprese



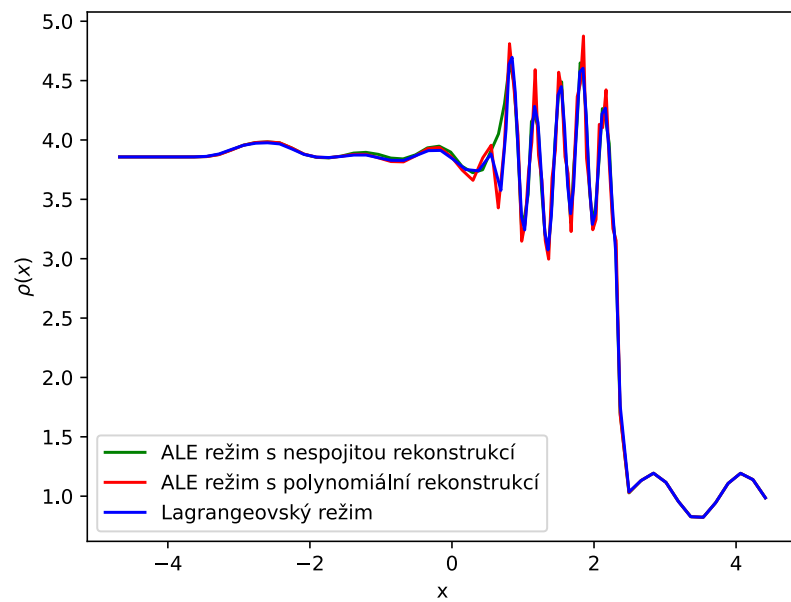
Obrázek 18: Výsledky numerického řešení Sodova problému, problém jsme řešili v ALE režimu a používali jsme nespojitou rekonstrukci s n buňkami

7.2 Shu-Osherův problém

Počáteční podmínky opět nalezneme v tabulce 5. V Shu-Osher problému obsahuje tedy menší hustota oscilace.

Okrajovou podmínkou na levé straně výpočetní oblasti je píst s konstantní rychlostí $u = 2.629369$, látka je tedy stlačována.

Na obrázku 19 můžeme vidět hustotu, poté co jsme problém simulovali za pomoci různých námi implementovaných metod.



Obrázek 19: Výsledky numerického řešení Shu-Osherova problému, problém jsme řešili na síťce s 80 buňkami

Závěr

V této práci jsme se zaměřili na cell-centered Eulerovsko-Lagrangeovské metody. Hlavním tématem práce bylo zkoumání Eulerovského kroku, konkrétně remapu a rekonstrukci veličin.

Popsali jsme existující polynomiální metody pro rekonstrukci veličin z diskrétních hodnot. Zabývali jsme se jak jednomateriálovými, tak multimateriálovými metodami. Oba dva typy metod jsme následně otestovali v námi implementovaném kódu za pomoci testů převzatých z odborné literatury. Po porovnání výsledků jsme zjistili, že naše implementace dosahuje podobných výsledků. Následně jsme se zabývali nepolynomiálními metodami pro rekonstrukci. Popsali jsme metodu THINC a následně jsme odvodili nově navrženou metodu pro rekonstrukci za pomoci nespojitě funkce. Tyto metody jsme následně porovnali s metodami polynomiálními. Zjistili jsme, že naše metoda dosahuje podobných výsledků jako metoda THINC i přes to, že její formulace je výrazně jednodušší.

Všechny tyto metody jsme poté použili v námi implementovaném 1D ALE kódu. Tímto kódem jsme následně simulovali několik problémů z hydrodynamiky.

Budoucí práce by se mohla zabývat především zobecněním THINC a nespojitě rekonstrukce do více dimenzí. Zvláště zajímavá by byla aproximace zakřivených materiálových rozhraní, což klasické metody pro materiálovou rekonstrukci neumí. Nová metoda by pak mohla rozhraní mezi materiály reprezentovat přesněji než při použití multimateriálových metod.

Dále by bylo možné zkoumat použití nepolynomiálních rekonstrukcí v případě staggered ALE metod.

Reference

- [1] N. R. Morgan and B. J. Archer. On the origins of lagrangian hydrodynamic methods. *Nuclear Technology*, 207(sup1):S147–S175, December 2021.
- [2] J. K. Dukowicz and J. R. Baumgardner. Incremental remapping as a transport/advection algorithm. *Journal of Computational Physics*, 160(1):318–335, May 2000.
- [3] L.G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics*, 184(1):266–298, January 2003.
- [4] M. Kucharik, M. Shashkov, and B. Wendroff. An efficient linearity-and-bound-preserving remapping method. *Journal of Computational Physics*, 188(2):462–471, July 2003.
- [5] J. Velechovsky, J. Breil, and R. Liska. Flux corrected remapping using piecewise parabolic reconstruction for 2D cell-centered ALE methods: PIECEWISE PARABOLIC RECONSTRUCTION FOR 2D CELL-CENTERED ALE METHODS. *International Journal for Numerical Methods in Fluids*, 76(9):575–586, November 2014.
- [6] J. Cheng and CW. Shu. A high order accurate conservative remapping method on staggered meshes. *Applied Numerical Mathematics*, 58(7):1042–1060, July 2008.
- [7] S. Galera, PH Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics*, 229(16):5755–5787, August 2010.
- [8] M. Berndt, J. Breil, S. Galera, M. Kucharik, PH Maire, and M. Shashkov. Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics*, 230(17):6664–6687, July 2011.
- [9] M. Kucharik and M. Shashkov. Conservative multi-material remap for staggered multi-material Arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics*, 258:268–304, February 2014.
- [10] F. Xiao, Y. Honma, and T. Kono. A simple algebraic interface capturing scheme using hyperbolic tangent function. *International Journal for Numerical Methods in Fluids*, 48(9):1023–1040, July 2005.
- [11] Z. Sun, S. Inaba, and F. Xiao. Boundary Variation Diminishing (Bvd) reconstruction: A new approach to improve Godunov schemes. *Journal of Computational Physics*, 322:309–325, October 2016.
- [12] K. Shyue and F. Xiao. An Eulerian interface sharpening algorithm for compressible two-phase flow: The algebraic THINC approach. *Journal of Computational Physics*, 268:326–354, July 2014.

- [13] M. Kucharik and R. Loubère. High-accurate and robust conservative re-mapping combining polynomial and hyperbolic tangent reconstructions. *Computers & Fluids*, 208:104614, August 2020.
- [14] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the Multi-dimensional Optimal Order Detection (Mood) on unstructured meshes with very high-order polynomials. *Computers & Fluids*, 64:43–63, July 2012.
- [15] F. Vilar. *A high-order Discontinuous Galerkin discretization for solving two-dimensional Lagrangian hydrodynamics*. Numerical Analysis, Université Sciences et Technologies - Bordeaux I, December 2012.
- [16] K. Michalak and C. Ollivier-Gooch. Limiters for unstructured higher-order accurate solutions of the euler equations. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2008. American Institute of Aeronautics and Astronautics.
- [17] T. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Sciences Meeting*, Reno, NV, U.S.A., January 1989. American Institute of Aeronautics and Astronautics.
- [18] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. In *31st Aerospace Sciences Meeting*, Reno, NV, U.S.A., January 1993. American Institute of Aeronautics and Astronautics.
- [19] C. F. Ollivier-Gooch. Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6–17, 1997.
- [20] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144(1):194–212, July 1998.
- [21] T. Barth and P. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. In *28th Aerospace Sciences Meeting*, Reno, NV, U.S.A., January 1990. American Institute of Aeronautics and Astronautics.
- [22] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):5361–5384, May 2008.
- [23] PH Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228(7):2391–2425, April 2009.
- [24] S Godunov, A Zabrodine, M Ivanov, A Kraiko, and G Prokopov. Resolution numérique des problèmes multidimensionnels de la dynamique des gaz, tions mir. *Moscow, Russia*, 1979.

- [25] R. Piessens, E. de Doncker-Kapenga, C.W. Überhuber, and D.K. Kahaner. *QUADPACK: a subroutine package for automatic integration*. Number 1 in Springer series in computational mathematics. Springer, Berlin Heidelberg, 1983.
- [26] L.G. Margolin and M. Shashkov. Second-order sign-preserving remapping on general grids. Technical report LA-UR-02-525, Los Alamos National Laboratory, 2002.