

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra mechaniky

DIPLOMOVÁ PRÁCE

Využití metody adaptivního síťování pro
numerické řešení problémů proudění tekutin

Plzeň 2019

Jan Basl

TADY BUDE ZADÁNÍ

Prohlášení

Prohlašuji, že jsem diplomovou práci na téma Využití metody adaptivního síťování pro numerické řešení problémů proudění tekutin vypracoval samostatně a s použitím uvedené literatury a pramenů.

V Plzni, 2019

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé diplomové práce, doc. Ing. Janu Vimmrovi, Ph.D. za věnovaný čas a užitečné rady při psaní této práce. Dále bych chtěl poděkovat mé manželce a rodině za psychickou podporu, nejen během psaní této práce.

Abstrakt

Diplomová práce se zabývá základy adaptivního síťování využitelného při numerickém řešení problémů proudění tekutin. V práci je aplikována metoda konečných diferencí v případě 1D problémů a metoda konečných objemů formulována na nestrukturované čtyřúhelníkové síti v případě 2D problémů. Pro řešení 2D problémů proudění je použit matematický model popsáný nelineárním systémem Eulerových rovnic. Pro aproximaci nevazkých numerických toků je implementováno explicitní AUSM schéma prvního řádu přesnosti. Časová diskretizace je provedena pomocí dvoustupňové Rungeovy-Kuttovy metody.

Hlavní částí práce je popis implementace algoritmů pro adaptaci sítě. V první části práce je popsán a implementován přístup k adaptaci typu „r-refinement“, který zachovává počet uzlů výpočetní sítě a pohybuje s uzly sítě, pro řešení 1D úloh. V druhé části práce je implementován přístup pro adaptaci výpočetní sítě typu „h-refinement“, který mění propojení jednotlivých buněk sítě, pro řešení 2D problémů proudění. Vyvinuté algoritmy jsou v práci testovány na vybraných testovacích úlohách.

Klíčová slova: metoda konečných objemů, metoda konečných diferencí, systém Eulerových rovnic, adaptace sítě, AUSM schéma

Abstract

Diploma thesis presents basics of the adaptive mesh refinement methods, focused on solving numerical problems of the fluid flow. In this thesis, finite difference method is used to solve 1D cases and finite volume method is used to solve 2D cases. To solve 2D cases of the fluid flow, mathematical model of non-linear inviscid fluid flow is implemented, described by Euler's system of equations. Inviscid numerical fluxes are approximated using explicit first order AUSM scheme. Time discretization is done by using two stage Runge-Kutta method.

Main focus of the thesis is to describe implementation of the algorithms used for the mesh refinement. First part of the thesis describes implementation of the r-refinement type adaptation, which maintains number of nodes and moves them, to solve 1D cases. In second part of the thesis, h-refinement type of adaptation is implemented, which increases number of nodes and changes their connections, to solve 2D cases. Developed algorithms are tested on selected test cases.

Key words: finite volume method, finite difference method, system of Euler's equations, adaptive mesh refinement, AUSM scheme

Obsah

Úvod	7
1 Shrnutí přístupů k adaptivním sítím	10
1.1 Přístup typu „h-refinement“	10
1.2 Přístup typu „p-refinement“	11
1.3 Přístup typu „r-refinement“	11
2 Adaptace výpočetní sítě v 1D	13
2.1 Princip ekvidistribuce	13
2.2 De Boorův algoritmus	14
2.3 Metoda BVP	15
2.4 Metody pohyblivé sítě	16
2.5 Numerické řešení adaptivních sítí v 1D	19
2.5.1 Porovnání popsaných algoritmů pro danou funkci hustoty sítě	19
2.5.2 Numerické řešení modelové nelineární vazké Burgerovy rovnice v 1D s využitím adaptace sítě	23
3 Numerické řešení proudění stlačitelných neviskózních a tepelně nevodivých tekutin ve 2D	33
3.1 Metoda konečných objemů	34
3.2 Metoda časové integrace	37
3.3 Explicitní AUSM schéma	39
4 Adaptace výpočetní sítě ve 2D	42
4.1 Implementace algoritmu pro adaptaci výpočetní sítě	42
4.1.1 Úpravy řešiče pro adaptaci sítě	44
4.1.2 Volba buněk určených pro adaptaci	46
4.1.3 Tvorba adaptované sítě	48

4.2	Numerické řešení testovacího příkladu: 2D GAMM kanál	49
4.2.1	Adaptace 2D GAMM kanálu	51
4.2.2	Numerické výsledky testovacího příkladu GAMM kanálu	51
	Závěr	58

Úvod

V problematice numerického řešení proudění tekutin je často problém zajistit dostatečnou kvalitu výpočetní sítě. V určitých místech výpočtové oblasti je potřeba velmi jemná výpočetní síť, v jiných částech výpočtové oblasti naopak postačuje poměrně hrubá síť. Z tohoto důvodu začali vznikat algoritmy a metody, které se snaží upravovat síť během výpočtu. Motivací pro vznik této práce bylo zejména proniknout do problematiky adaptace sítě a využít ji pro řešení problémů zachycení rázových vln, se kterou se autor setkal při tvorbě své bakalářské práce.

Předkládaná diplomová práce se zabývá základy matematického modelování proudění tekutin s využitím metod a algoritmů adaptivních sítí. Cílem práce je zejména popsat metodiku a programovou implementaci problému adaptace sítě. K implementaci je využito výpočtové prostředí MATLAB. V práci jsou také popsány problémy, kterými je potřeba se zabývat při implementaci adaptivních algoritmů.

Diplomová práce je rozdělena do čtyř kapitol. V první kapitole jsou shrnuty přístupy k adaptaci sítě a příklady jejich využití. Jedná se o přístup typu „h-refinement“, „p-refinement“ a „r-refinement“.

Druhá kapitola je zaměřena na algoritmy pro adaptaci výpočetní sítě typu „r-refinement“, kdy je zachováván počet uzlů sítě a dochází k pohybu uzlů. V této kapitole je popsán základní matematický model, popisující pohyb sítě. Dále jsou v kapitole popsány některé vlastnosti a problémy spojené s řešením problematiky adaptivního síťování pomocí typu „r-refinement“. Druhá část kapitoly je věnována porovnání jednotlivých algoritmů pro adaptaci sítě. Poslední část kapitoly pak obsahuje numerické řešení modelové nelineární vazké Burgersovy rovnice v 1D pomocí metody konečných diferencí s využitím implementovaného adaptivního algoritmu typu „r-refinement“.

Třetí kapitola je věnována popisu numerického řešení nelineárního systému Eulerových rovnic. Pro prostorovou diskretizaci nelineárního systému Eulerových rovnic byla zvolena metoda konečných objemů ve 2D formulovaná pro strukturované čtyřúhelníkové síť. Pro aproximaci ne vazkých numerických toků je v této práci využito explicitní AUSM schéma, které je prvního řádu přesnosti. Pro diskretizaci v čase je pak využita Rungeova-Kuttova metoda druhého řádu.

Čtvrtá a poslední kapitola této diplomové práce se věnuje přístupu k adaptaci typu „h-refinement“ ve 2D. Tato kapitola je zaměřena na programovou realizaci tohoto přístupu. V kapitole jsou popsány problémy při implementaci, potřebné úpravy výchozího řešiče a převod ze strukturované čtyřúhelníkové sítě na nestrukturovanou čtyřúhelníkovou síť. Jsou zde také uvedeny některé problémy, které souvisí s adaptací sítě. V druhé části této kapitoly je pak navržený a implementovaný algoritmus využit pro řešení úlohy proudění stlačitelné nevazké a tepelně nevodivé tekutiny ve známém testovacím GAMM kanálu ve 2D.

V závěru práce je poté provedeno celkové shrnutí a zhodnocení vyvinutých algoritmů pro adaptaci sítě.

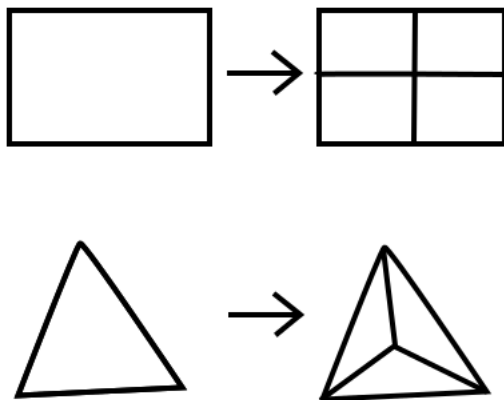
1 Shrnutí přístupů k adaptivním sítím

Při numerickém řešení úloh, nejen v oblasti proudění tekutin, je velmi často problém dosáhnout optimální kvality sítě, pro dosažení co nejpřesnějších výsledků. Pro kvalitní řešení je také velmi často potřeba velkého množství buněk sítě, čímž dochází k nárůstu výpočetního času. Z těchto důvodů začaly vznikat metody, které pomáhají optimalizovat výpočetní síť tak, aby bylo dosaženo co nejlepšího poměru přesnosti výsledků a časové náročnosti výpočtu. Tyto metody lze obecně nazvat adaptací výpočetní sítě, neboť dochází k přizpůsobení (adaptaci) dané sítě v závislosti na zvoleném parametru, funkci nebo normě.

Adaptivní sítě v oblasti proudění tekutin nacházejí uplatnění například v modelování vícefázového proudění [6], interakci tekutiny s pevnými částicemi [7] nebo pro zachycení nespojitostí typu rázových vln, kdy je potřeba zajistit lokální zahuštění sítě v oblasti vzniku rázové vlny. Postup při adaptaci sítě lze rozdělit na tři základní přístupy [3]. Cílem všech níže zmíněných metod je zvýšit kvalitu sítě, respektive tedy numerického řešení. Každá metoda využívá různých přístupů a postupů, jak zvýšení kvality dosáhnout.

1.1 Přístup typu „h-refinement“

Při využití přístupu typu „h-refinement“ dochází ke změně propojení jednotlivých buněk. Zpravidla to znamená, že dochází ke změně počtu buněk, respektive uzlů sítě [1]. Základním přístupem je rozdělení buněk na buňky menší, lze také naopak buňky slučovat. Hlavní výhodou tohoto přístupu je zachování topologie sítě a relativně snadná implementace, v porovnání s ostatními metodami. Nevýhodou metody je poměrně rychlý nárůst počtu buněk a tím pádem nárůst výpočetního času [4]. Pokud budeme uvažovat 2D úlohu, poté základní přístup k rozdělení buněk může vypadat například tak, jak je zobrazeno na následujícím obrázku.



Obr. 1: Princip metody typu „h-refinement“

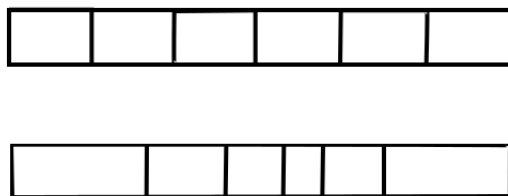
1.2 Přístup typu „p-refinement“

Přístup „p-refinement“ nachází uplatnění zejména v metodě konečných prvků (*finite element method*) [2] a pro metodu konečných objemů (*finite volume method*) nemá větší význam. Metoda je postavená na zvyšování řádu přesnosti v uzlech sítě, respektive tedy zvyšování řádu polynomu. Vzhledem k neuplatnitelnosti pro metodu konečných objemů, využití v této práci, je zde tento přístup zmíněn jen pro úplnost.

1.3 Přístup typu „r-refinement“

Poslední metoda, „r-refinement“, zachovává počet buněk sítě, respektive uzlů [5]. Pro zvýšení přesnosti řešení dochází k posouvání uzlů sítě do regionů, kde dochází k dějům vyžadujícím zahuštění sítě (např. výše zmíněná rázová vlna). K sledování a určení směru pohybu uzlů se využívá různých, matematicky definovaných monitorovacích funkcí. To zpravidla znamená, že do stávajícího modelu zadaného problému je potřeba přidat další, z pravidla diferenciální, rovnice a vhodně tyto nové rovnice propojit se systémem stávajících rovnic. Tento přístup je velmi vhodný pro řešení 1D problémů, kde je možné poměrně dobře propojit systém rovnic s rovnicí pro adaptaci sítě. Ve více dimenzích

dochází ke značné komplikaci v implementaci rovnic a zajištění stability řešení. Dalším problémem je kvalita sítě, kterou je potřeba velmi pečlivě hlídat. Výhodou metody je menší počet buněk, tedy teoreticky rychlejší výpočet. Je potřeba ale brát v úvahu, že výpočet je zpomalován nutností přidat další rovnice do již stávajícího systému rovnic. Příklad adaptace typu „r-refinement“ je zobrazen na následujícím obrázku.



Obr. 2: Princip metody typu „r-refinement“

V této práci je využito prvního přístupu pro řešení 2D úloh s využitím metody konečných objemů a třetího přístupu pro řešení 1D úloh s využitím metody konečných diferencí.

2 Adaptace výpočetní sítě v 1D

Problém adaptivní sítě lze slovy definovat například tak, že máme funkci $f(x)$ na omezeném intervalu a máme tuto funkci aproximovat v konečném počtu bodů N . Jak daleko od sebe zvolit jednotlivé body vytvořené sítě?

Zvolíme-li vhodně nějakou funkci hustoty sítě $\rho(x)$, která představuje jakousi chybu numerické aproximace, potom v místech, kde tato funkce nabývá vysokých hodnot, bude vzdálenost mezi body výpočetní sítě menší. Naopak tam kde jsou hodnoty funkce $\rho(x)$ nízké, budou vzdálenosti mezi body výpočetní sítě větší. Tento princip se nazývá princip ekvidistribuce. Tento princip a základní metody používané pro řešení adaptace sítě v 1D jsou popsány v této kapitole a vycházejí z literatury [8], [9] [12].

2.1 Princip ekvidistribuce

Uvažujme funkci $\rho = \rho(x) > 0$ na omezeném intervalu $\langle a, b \rangle$. Budeme hledat takovou síť $\tau_h : x_1 = a < x_2 \cdots < x_N = b$, která přerozdělí funkci ρ na podintervaly definované pomocí jednotlivých bodů sítě tak, aby platilo

$$\int_{x_1}^{x_2} \rho(x) dx = \cdots = \int_{x_{N-1}}^{x_N} \rho(x) dx. \quad (2.1)$$

Tuto rovnici lze slovně popsat tak, že plocha pod funkcí $\rho(x)$ je stejná pro každý podinterval. Funkci $\rho(x)$ budeme nazývat funkce hustoty sítě (*mesh density function*). Kvadrátu této funkce $\rho(x)^2$ budeme říkat monitorovací funkce (*monitor function*).

Dále uvažujme, že síť τ_h je vytvořena pomocí transformace $x = x(\xi) : \langle 0, 1 \rangle \rightarrow \langle a, b \rangle$, tedy

$$x_i = x(\xi_i), j = 1, \dots, N, \quad (2.2)$$

$$\xi_i = \frac{i-1}{N-1}. \quad (2.3)$$

Princip ekvidistribuce lze tedy zapsat jako

$$\int_a^{x(\xi_i)} \rho(x) dx = \sigma \xi_i, \quad (2.4)$$

kde

$$\sigma = \int_a^b \rho(x) dx. \quad (2.5)$$

Obecně lze psát

$$\int_a^{x(\xi)} \rho(x) dx = \sigma \xi. \quad (2.6)$$

kde $\xi \in (0, 1)$. Derivací podle ξ lze ukázat, že platí

$$\rho(x) \frac{dx}{d\xi} = \sigma. \quad (2.7)$$

Díky tomuto vztahu lze definovat inverzní transformaci $\xi = \xi(x) : \langle a, b \rangle \rightarrow \langle 0, 1 \rangle$

$$\frac{1}{\rho(x)} \frac{d\xi}{dx} = \frac{1}{\sigma}. \quad (2.8)$$

2.2 De Boorův algoritmus

Jednoduchou metodou, kterou je možné použít pro hledání ekvidistribuované sítě je de Boorův algoritmus. Uvažujeme opět funkci hustoty sítě $\rho(x)$ a nějakou dočasnou síť $\tau_b : \tilde{x}_1 = a < \tilde{x}_2 < \dots < \tilde{x}_K = b$. Tuto síť lze chápat jako síť v dané časové hladině, respektive iteraci. Funkci $\rho(x)$ aproximujeme jako po částech konstantní funkci $p(x)$

$$p(x) = \begin{cases} \frac{[\rho(\tilde{x}_1) + \rho(\tilde{x}_2)]}{2} & x \in \langle \tilde{x}_1, \tilde{x}_2 \rangle \\ \frac{[\rho(\tilde{x}_2) + \rho(\tilde{x}_3)]}{2} & x \in \langle \tilde{x}_2, \tilde{x}_3 \rangle \\ \vdots & \vdots \\ \frac{[\rho(\tilde{x}_{K-1}) + \rho(\tilde{x}_K)]}{2} & x \in \langle \tilde{x}_{K-1}, \tilde{x}_K \rangle \end{cases}. \quad (2.9)$$

Dále řešíme problém ekvidistribuce, ale místo funkce $\rho(x)$ máme nyní funkci $p(x)$, která je po částech konstantní. Z principu ekvidistribuce zapíšeme

$$P(x) = \int_a^x p(x) dx, \quad (2.10)$$

což lze přepsat pomocí sumace jako

$$P(\tilde{x}_j) = \sum_{i=2}^j (\tilde{x}_i - \tilde{x}_{i-1}) \frac{[\rho(\tilde{x}_i) + \rho(\tilde{x}_{i-1})]}{2}, j = 2, 3, \dots, K. \quad (2.11)$$

Z rovnice (2.4) potom plyne

$$P(x_j) = \xi_j P(b), j = 2, 3, N - 1, \quad (2.12)$$

kde $P(b) = P(\tilde{x}_K)$. Pro určení x_j je potřeba nejprve určit index k tak, aby platilo

$$P(\tilde{x}_{k-1}) < \xi_j P(b) < P(\tilde{x}_k). \quad (2.13)$$

Díky tomu, že $P(x)$ je po částech lineární, neboť se jedná o integraci po částech konstantní funkce, lze x_j určit přímo jako

$$(x_j - \tilde{x}_{k-1}) \frac{[\rho(\tilde{x}_{k-1}) + \rho(\tilde{x}_k)]}{2} = \xi_j P(b) - P(\tilde{x}_{k-1}), \quad (2.14)$$

resp. z této rovnice lze explicitně vyjádřit x_j jako

$$x_j = \tilde{x}_{k-1} + \frac{2 [\xi_j P(b) - P(\tilde{x}_{k-1})]}{\rho(\tilde{x}_{k-1}) + \rho(\tilde{x}_k)}. \quad (2.15)$$

Vzhledem k tomu, že funkce $p(x)$ je po částech konstantní, je pro dosažení řešení potřeba tuto rovnici iterovat.

2.3 Metoda BVP

De Boorův algoritmus je poměrně jednoduchý a snadno algoritmizovatelný, nicméně je velmi obtížné rozšířit ho do více dimenzí. Proto jako další uvedeme metodu pro výpočet ekvidistantní sítě pomocí okrajové úlohy (*BVM - Boundary value problem method*). Nejprve zderivujeme rovnici (2.7) podle proměnné ξ

$$\frac{dx}{d\xi} \left(\rho(x) \frac{dx}{d\xi} \right) = 0 \quad (2.16)$$

a nadále platí okrajové podmínky $x(0) = a$, $x(1) = b$.

Úlohu hledání sítě splňující parametry zmíněné výše je vhodné formulovat pomocí variačního počtu následujícím způsobem.

Hledejme transformaci $x = x(\xi)$ která minimalizuje funkcionál

$$I(x) = \frac{1}{2} \int_0^1 \left[\rho(x) \frac{dx}{d\xi} \right] d\xi. \quad (2.17)$$

Tuto úlohu lze také formulovat pro inverzní transformaci $\xi = \xi(x)$.

$$I(\xi) = \frac{1}{2} \int_a^b \frac{1}{\rho(x)} \left(\frac{d\xi}{dx} \right)^2 dx. \quad (2.18)$$

Předpokládejme dále, že je dána funkce ρ^n na síti τ_h^n . Rovnici (2.16) diskretizujeme pomocí centrálních diferencí následujícím způsobem

$$\frac{2}{\xi_{i+1} - \xi_{i-1}} \left[\frac{\rho(x_{i+1}^n) + \rho(x_i^n)}{2} \cdot \frac{(x_{i+1}^{n+1} - x_i^{n+1})}{(\xi_{i+1} - \xi_i)} - \frac{\rho(x_i^n) + \rho(x_{i-1}^n)}{2} \cdot \frac{(x_i^{n+1} - x_{i-1}^{n+1})}{(\xi_i - \xi_{i-1})} \right] = 0. \quad (2.19)$$

2.4 Metody pohyblivé sítě

Výše zmíněná metoda uvažuje, že $\rho \neq \rho(t)$. Při řešení časově závislých problémů se ale funkce hustoty sítě bude měnit s časem, je tedy potřeba zavést metodu generování časově závislé sítě. Vyjdeme z rovnice (2.16), ovšem nyní bude platit $\rho = \rho(x, t)$

$$\frac{\partial x}{\partial \xi} \left(\rho(x, t) \frac{\partial x}{\partial \xi} \right) = 0 \quad (2.20)$$

s platnostní okrajové podmínky $x(0) = a$, $x(1) = b$. Pro další postup je vhodné dostat rovnici do tvaru, ve které se explicitně vyskytuje rychlost změny sítě (*mesh speed*) $\frac{\partial x}{\partial t}$. Semidiskretizace takové rovnice totiž povede na systém obyčejných diferenciálních rovnic. Diskretizace (2.20) povede na systém algebraických rovnic. Při úplné diskretizaci by pak ve druhém zmíněném případě vznikl systém algebro-diferenciálních rovnic, který je obecně obtížně řešitelný v porovnání se systémem obyčejných diferenciálních rovnic.

Rovnice zahrnující rychlost změny sítě budeme nazývat MMPDE (*Moving mesh partial differential equation*). Způsobů, jak definovat tyto rovnice je celá řada a existují různé tvary těchto rovnic. V této práci budeme MMPDE volit jako *Gradient flow equation* zvoleného funkcionalu

$$I(\xi) = \frac{1}{2} \int_a^b \frac{1}{\rho(x, t)} \left(\frac{\partial \xi}{\partial x} \right)^2 dx. \quad (2.21)$$

Pro směr ξ , který snižuje hodnotu $I(\xi)$ se jedná o rovnici

$$\frac{\partial \xi}{\partial t} = \frac{P}{\tau} \frac{\delta I}{\delta \xi}. \quad (2.22)$$

Zde $\tau > 0$ je volená konstanta, kterou lze chápat jako doba odezvy sítě na změny $\rho(x, t)$ a P je diferenciální operátor.

Lze ukázat [8], že platí

$$\frac{\delta I}{\delta \xi} = -\frac{\partial}{\partial x} \left(\frac{\partial \xi}{\rho \partial x} \right). \quad (2.23)$$

Tedy můžeme přepsat rovnici (2.22) do tvaru

$$\frac{\partial \xi}{\partial t} = -\frac{P}{\tau} \frac{\partial}{\partial x} \left(\frac{\partial \xi}{\rho \partial x} \right). \quad (2.24)$$

Tato rovnice je definována jako transformace $\xi = \xi(x, t)$. Jak již bylo naznačeno výše, je vhodnější zaměnit proměnné tak aby $x = x(\xi, t)$, neboť původní varianta explicitně neurčuje polohu bodu ve „fyzikálním“ prostoru. Změnu proměnných lze provést následujícím způsobem

$$\xi = \xi(x(\xi, t), t). \quad (2.25)$$

Derivováním tohoto výrazu podle ξ

$$1 = \frac{\partial \xi}{\partial x} \frac{\partial x}{\partial \xi}, \quad (2.26)$$

respektive toto lze zapsat jako

$$\left(\frac{\partial \xi}{\partial x} \right)^{-1} = \frac{\partial x}{\partial \xi}. \quad (2.27)$$

Dále derivováním podle t

$$0 = \frac{\partial \xi}{\partial t} + \frac{\partial \xi}{\partial x} \frac{\partial x}{\partial t}, \quad (2.28)$$

respektive

$$\frac{\partial x}{\partial t} = -\frac{\partial x}{\partial \xi} \frac{\partial \xi}{\partial t}. \quad (2.29)$$

S využitím těchto vztahů lze (2.22) přepsat do tvaru

$$\frac{\partial x}{\partial t} = \frac{1}{\tau} \frac{\partial x}{\partial \xi} P \left(\rho \frac{\partial x}{\partial \xi} \right)^{-2} \left(\frac{\partial x}{\partial \xi} \right)^{-1} \frac{\partial}{\partial \xi} \left(\rho \frac{\partial x}{\partial \xi} \right). \quad (2.30)$$

Typ MMPDE specifikujeme vhodnou volbou P . Například pro $P = (\rho x_\xi)^2$ dostáváme MMPDE5

$$\frac{\partial x}{\partial t} = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left(\rho \frac{\partial x}{\partial \xi} \right). \quad (2.31)$$

Jinou volbou P , například $P = \frac{(\rho x_\xi)^2}{\rho}$ dostáváme modifikovanou MMPDE5

$$\frac{\partial x}{\partial t} = \frac{1}{\tau \rho} \frac{\partial}{\partial \xi} \left(\rho \frac{\partial x}{\partial \xi} \right). \quad (2.32)$$

Tato rovnice může být diskretizována například pomocí metody konečných diferencí s využitím centrálních diferencí

$$\frac{dx_i}{dt} = \frac{1}{\rho_i \tau \Delta \xi^2} \left[\frac{1}{2} (\rho_{i+1} + \rho_i) (x_{i+1} - x_i) - \frac{1}{2} (\rho_i + \rho_{i-1}) (x_i - x_{i-1}) \right]. \quad (2.33)$$

Časovou derivaci na levé straně rovnice lze aproximovat například pomocí zpětné diferenční formule

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = \frac{1}{\rho_i^n \tau \Delta \xi^2} \left[\frac{1}{2} (\rho_{i+1}^n + \rho_i^n) (x_{i+1}^{n+1} - x_i^{n+1}) - \frac{1}{2} (\rho_i^n + \rho_{i-1}^n) (x_i^{n+1} - x_{i-1}^{n+1}) \right]. \quad (2.34)$$

Další možností jak volit diferenciální operátor P je například

$$P = -\frac{\partial x}{\partial \xi} \left(\frac{\partial}{\partial \xi} \rho \frac{\partial}{\partial \xi} \right)^{-1} \left(\rho \frac{\partial x}{\partial \xi} \right)^2 \frac{\partial x}{\partial \xi}. \quad (2.35)$$

Tato volba vede na rovnici MMPDE4

$$\frac{\partial}{\partial \xi} \left(\rho \frac{\partial x_t}{\partial \xi} \right) = -\frac{1}{\tau} \frac{\partial}{\partial \xi} \left(\rho \frac{\partial x}{\partial \xi} \right). \quad (2.36)$$

Společným znakem všech MMPDE rovnic je člen

$$\frac{1}{\tau} \frac{\partial}{\partial \xi} \left(\rho \frac{\partial x}{\partial \xi} \right). \quad (2.37)$$

Tento člen uvádí síť do pohybu a zároveň udržuje princip ekvidistribuce v platnosti. Tento člen vymizí, pokud je princip ekvidistribuce splněn, tedy když je síť přerozdělena, ustane pohyb sítě.

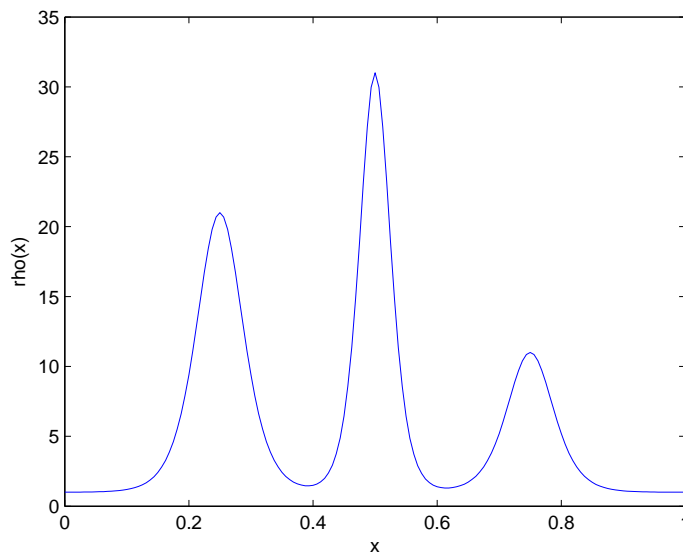
2.5 Numerické řešení adaptivních sítí v 1D

V předchozích kapitolách byly popsány tři metody pro tvorbu adaptivní sítě v 1D. Porovnejme tyto tři metody přístupu k adaptaci sítě na jednoduché úloze.

2.5.1 Porovnání popsaných algoritmů pro danou funkci hustoty sítě

Uvažujme funkci hustoty sítě ve tvaru

$$\begin{aligned} \rho(x) = & 1 + 10(1 - \tanh^2(10(x - 0.75))) + \\ & + 20(1 - \tanh^2(20(x - 0.25))) + 30(1 - \tanh^2(30(x - 0.5))). \end{aligned} \quad (2.38)$$

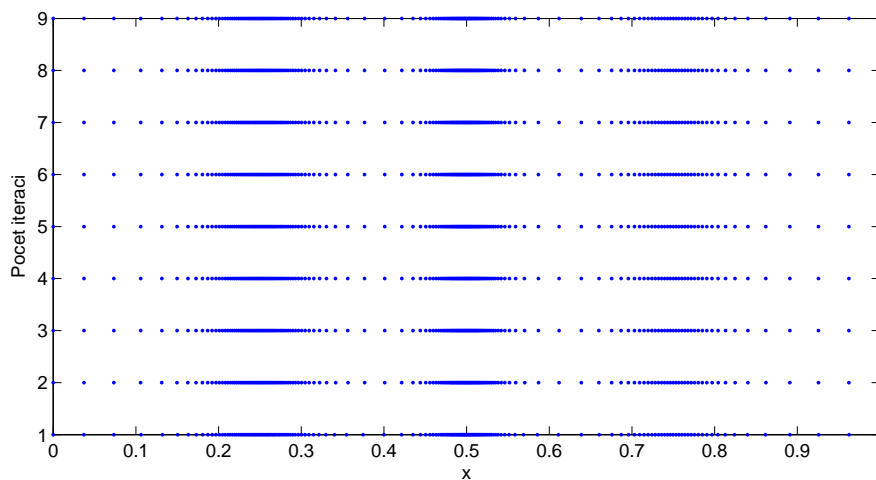


Obr. 3: Funkce hustoty sítě

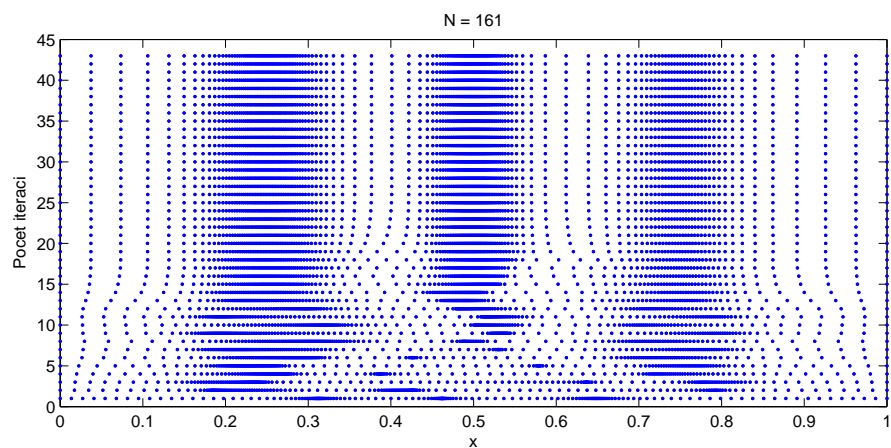
Na tuto funkci postupně aplikujeme výše popsané algoritmy pro tvorbu adaptivní sítě. Vždy přitom vycházíme z rovnoměrně rozložené sítě o 160 prvcích na intervalu $\langle 0, 1 \rangle$. Budeme přitom sledovat počet iterací potřebný k dosažení normy

$$\|\tau_h^{n+1} - \tau_h^n\| \equiv \|x(j)^{n+1} - x(j)^n\| < 10^{-8}, \quad (2.39)$$

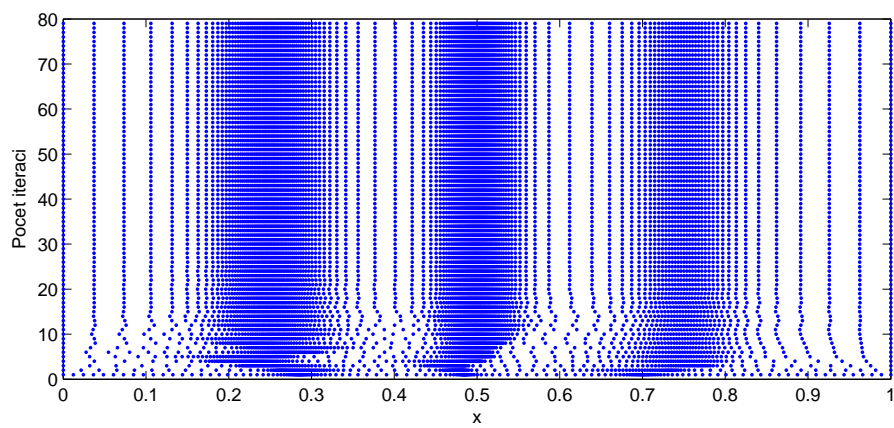
tedy výpočet se zastaví v momentě kdy rozdíl mezi rozložením sítě v následujících iteracích bude menší než 10^{-8} . Průběhy konvergence a průběhy změn polohy bodů sítě jsou zobrazeny na následujících obrázcích.



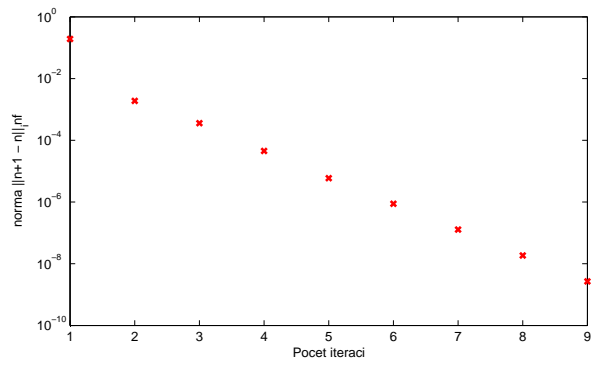
Obr. 4: Změna polohy bodů sítě pro de Boorův algoritmus



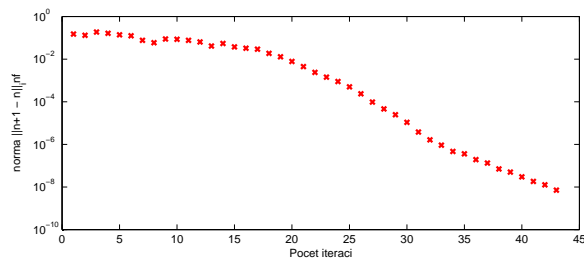
Obr. 5: Změna polohy bodů sítě pro BVP algoritmus



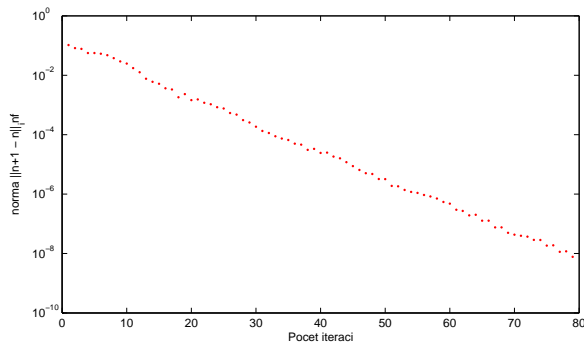
Obr. 6: Změna polohy bodů sítě pro MMPDE5 algoritmus



Obr. 7: Průběh konvergence pro de Boorův algoritmus

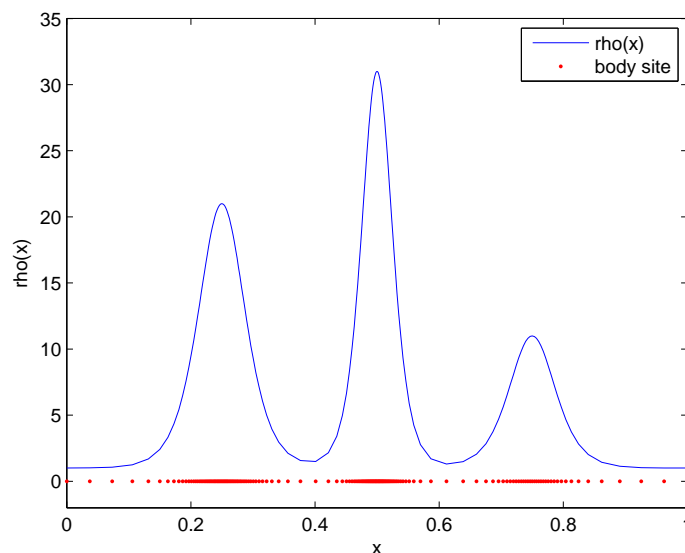


Obr. 8: Průběh konvergence pro BVP algoritmus



Obr. 9: Průběh konvergence pro MMPDE5 algoritmus

Jak je vidět z obrázků, síť se zahustila v místech, kde se nacházejí maxima funkce $\rho(x)$. Pro názornost ještě uveďme graf ve kterém je zobrazeno výsledné rozložení sítě a funkce $\rho(x)$ například pro algoritmus MMPDE5.



Obr. 10: $\rho(x)$ a rozložení sítě

Doba výpočtu všech algoritmů je přibližně stejná. Poznamenejme ještě, že konvergence MMPDE metody je ovlivněna volbou časového kroku Δt a volbou konstanty τ . Pro tuto konkrétní úlohu byly zvoleny hodnoty $\Delta t = 0.1$ a $\tau = 1$.

2.5.2 Numerické řešení modelové nelineární vazké Burgersovy rovnice v 1D s využitím adaptace sítě

Jako příklad složitější úlohy dále uveďme řešení vazké Burgersovy rovnice v 1D, která má tvar

$$u_t + \left(\frac{u^2}{2} \right) = \varepsilon u_{xx}, \quad (2.40)$$

kde $\varepsilon > 0$ představuje umělou vazkost. Budeme přitom uvažovat dva případy pro různé počáteční podmínky

$$u(0) = \sin(2\pi x) + \frac{1}{2} \sin(\pi x), \quad (2.41)$$

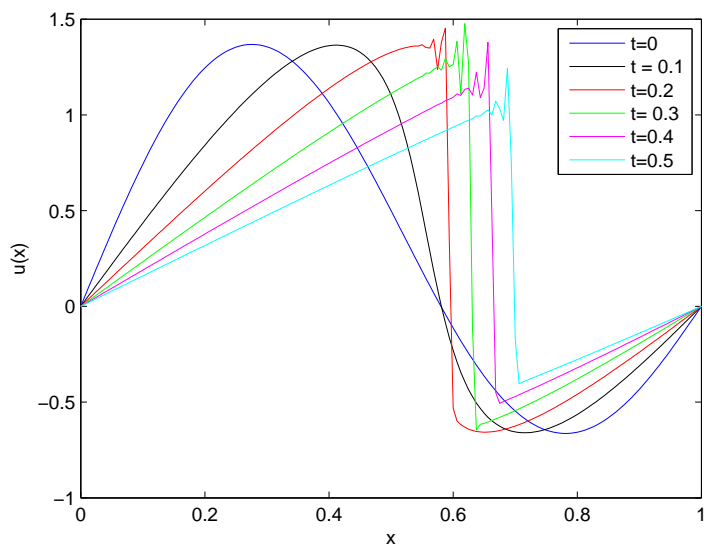
$$u(0) = \begin{cases} 1 & \text{pro } x \leq 0.5 \\ 0 & \text{pro } x > 0.5 \end{cases}. \quad (2.42)$$

Při řešení budeme uvažovat časový interval $t \in (0, 0.5)$ a umělou vazkost $\varepsilon = 10^{-3}$. Jako v předchozím případě budeme uvažovat síť o 160 prvcích. Pro řešení využijeme metodu konečných diferencí a využijeme například centrálních diferencí v prostoru a dopředné difference v čase

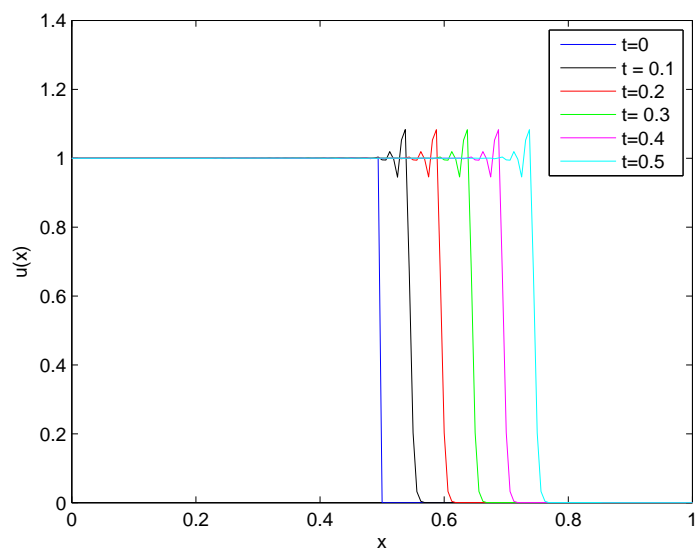
$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\varepsilon}{h^2} (u_{i+1} - 2u_i + u_{i-1}) - \frac{1}{4h} (u_{i+1}^2 - u_{i-1}^2), \quad (2.43)$$

kde $h = \frac{1}{N-1}$. Pro diskretizaci Burgersovy rovnice můžeme ale použít jakékoliv jiné stabilní schéma, viz například [10].

Při řešení Burgersovy rovnice na rovnoměrně rozložené síti nastávají oscilace v řešení v místech, kde vznikají nespojitosti typu rázové vlny. Řešení pro tuto síť je zobrazeno na následujících obrázcích. Z obrázků je patrný vznik oscilací v řešení v místě nespojitostí. Pro snížení oscilací by bylo potřeba zvýšit počet prvků sítě.



Obr. 11: Numerické řešení Burgersovy rovnice na rovnoměrně rozložené výpočetní síti pro počáteční podmínku (2.41)



Obr. 12: Numerické řešení Burgersovy rovnice na rovnoměrně rozložené výpočetní síti pro počáteční podmínku (2.42)

V dalších případech budeme uvažovat stejný počet uzlů, ale s využitím adaptace sítě metodou MMPDE, konkrétně pomocí modifikované MMPDE5 (2.32). Je potřeba si uvědomit, že nyní neřešíme pouze jednu rovnici pro tvorbu sítě, ale navíc přibyla rovnice pro řešení Burgersovy rovnice. K řešení přistupovat dvěma různými způsoby.

Popis implementace

Prvním způsobem jak přistupovat k danému problému je řešit rovnici pro adaptaci sítě současně s řešením parciální diferenciální rovnice. Zavedeme-li nový vektor neznámých

$$\mathbf{y} = [u_1(t), u_2(t), u_{N-1}(t), u_N(t), x_1(t), x_2(t), \dots, x_{N-1}(t), x_N(t)]^T, \quad (2.44)$$

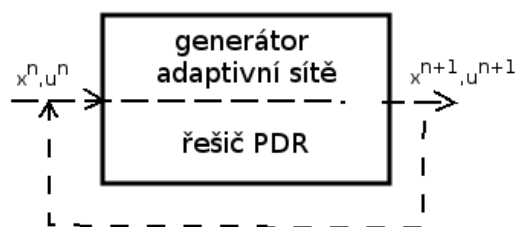
pak můžeme systém diferenciálních rovnic zahrnujících rovnici pro adaptaci sítě a parciální diferenciální rovnici (v našem případě Burgersovu) zapsat jako

$$\mathbf{f}(t, \mathbf{y}, \mathbf{y}') = \mathbf{0}. \quad (2.45)$$

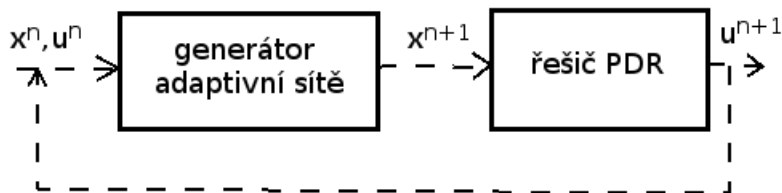
Hlavní výhodou tohoto přístupu je těsná souvislost sítě s řešením rovnice, tedy síť reaguje rychle na změny v řešení. Další výhodou je to, že v softwarovém prostředí MATLAB existuje řada implicitních řešičů (například ODE15i), které lze využít pro řešení tohoto systému. Nevýhodou tohoto přístupu je zejména to, že nově vzniklý systém bývá velmi často silně nelineární a má poměrně složitou strukturu. Nový systém rovnic také ztrácí některé důležité vlastnosti jako je symetrie nebo pozitivní definitnost. Systém je tedy obecně vzato mnohem složitější řešit, nicméně díky přímé závislosti sítě na řešení dostáváme kvalitnější výsledky než je tomu u přístupu, který si popíšeme nyní.

Druhým způsobem, jak přistupovat k adaptaci sítě je nejprve řešit rovnici pro adaptaci sítě a poté aplikovat nové hodnoty sítě na řešení dané parciální diferenciální rovnice. Zásadním problémem tohoto přístupu je fakt, že novou hladinu sítě x^{n+1} počítáme pomocí řešení z předchozí hladiny u^n . Vzniká tedy jakési zpoždění v řešení. Tento jev lze do jisté míry eliminovat volbou velmi malého časového kroku Δt . Další možností jak toto zpoždění minimalizovat je iterovat několikrát výpočet $n + 1$ hladiny sítě x^{n+1} . Zásadní výhodou tohoto přístupu je fakt, že od sebe lze odseparovat řešič sítě a řešič rovnice. Tedy lze vytvořit skript pro adaptaci sítě a přidat ho k již fungujícímu softwaru. Díky tomu je také možné ve skriptu pro adaptaci sítě provádět větší změny a volit různé typy a přístupy k adaptaci sítě.

První popsany přístup je kvůli složité implementaci omezený prakticky jen pro 1D případy. Druhý zmíněný případ je možný aplikovat ve více dimenzích, je tedy používán v různých aplikacích výrazně častěji, byť za cenu nižší přesnosti adaptivní sítě a možnosti vzniku nestabilit v řešení. Pro lepší názornost je na následujících obrázcích zobrazeno schéma obou přístupů.



Obr. 13: Schéma algoritmizace při současném řešení adaptace sítě a parciální diferenciální rovnice



Obr. 14: Schéma algoritmizace při odděleném řešení adaptace sítě a parciální diferenciální rovnice

Před samotným numerickým řešením je potřeba vhodně zvolit funkci hustoty sítě. Jedna z možností, jak tuto funkci volit je popsána v následujících odstavcích.

Volba funkce hustoty sítě

Existuje řada způsobů, jak volit funkci $\rho(x, t)$. Touto problematikou se zabývá celá kapitola v literatuře [8] a volba jednotlivých funkcí je zde popsána a matematicky zdůvodněna. Pro úlohu Burgersovy rovnice můžeme zvolit $\rho(x)$ například jako

$$\rho_i = \left(1 + \frac{1}{\alpha} |u_{xx,i}|^2\right)^{\frac{1}{3}}, \quad (2.46)$$

kde

$$\alpha = \max \left\{ 1, \left[\sum_{i=2}^N \left(\frac{x_i - x_{i-1}}{2} \right) \left(|u_{xx,i}|^{\frac{2}{3}} + |u_{xx,i-1}|^{\frac{2}{3}} \right) \right]^3 \right\}. \quad (2.47)$$

Druhé derivace vyskytující se v rovnici můžeme diskretizovat jako

$$u_{xx,1} = 2 \left[\frac{(x_2 - x_1)(u_3 - u_1) - (x_3 - x_1)(u_2 - u_1)}{(x_3 - x_1)(x_2 - x_1)(x_3 - x_2)} \right], \quad (2.48)$$

$$u_{xx,i} = \frac{2}{x_{i+1} - x_{i-1}} \left[\frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_i} - \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \right], \quad (2.49)$$

$$u_{xx,N} = 2 \left[\frac{(x_{N-1} - x_N)(u_{N-2} - u_N) - (x_{N-2} - x_N)(u_{N-1} - u_N)}{(x_{N-2} - x_N)(x_{N-1} - x_N)(x_{N-2} - x_{N-1})} \right]. \quad (2.50)$$

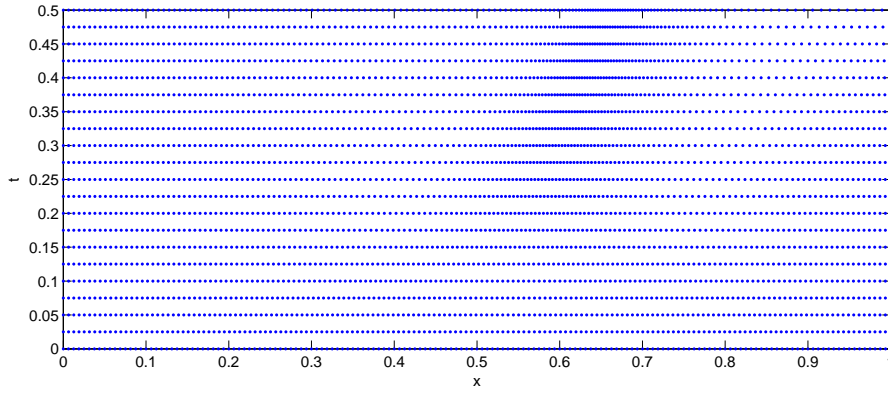
Při numerickém řešení se často používá funkce, která zpřesní $\rho(x, t)$. Tuto funkci lze implementovat například takto

$$\begin{cases} \rho_1 = \frac{1}{2} (\rho_1 + \rho_2), \\ \rho_i = \frac{1}{4} (\rho_{i-1} + 2\rho_i + \rho_{i+1}), \\ \rho_N = \frac{1}{2} (\rho_N + \rho_{N-1}). \end{cases} \quad (2.51)$$

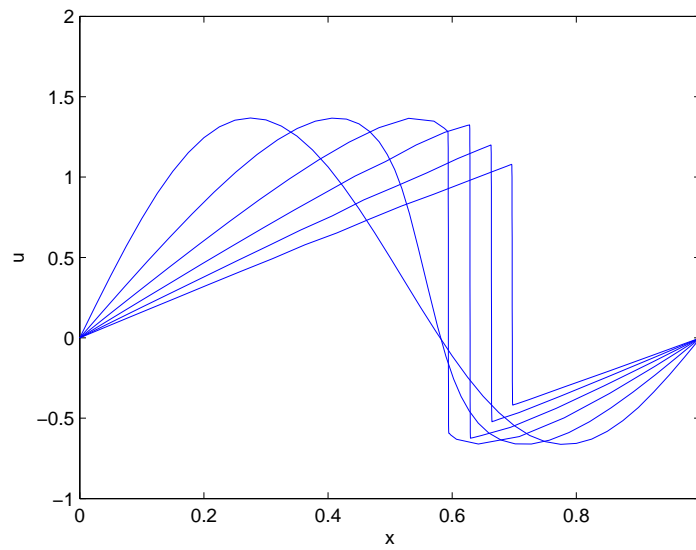
Toto schéma několikrát iterujeme (osvědčily se přibližně čtyři iterace). Takto připravenou funkci hustoty sítě $\rho(x, t)$ implementuje do rovnice (2.34).

Výsledky numerického řešení Burgersovy rovnice s využitím adaptace sítě

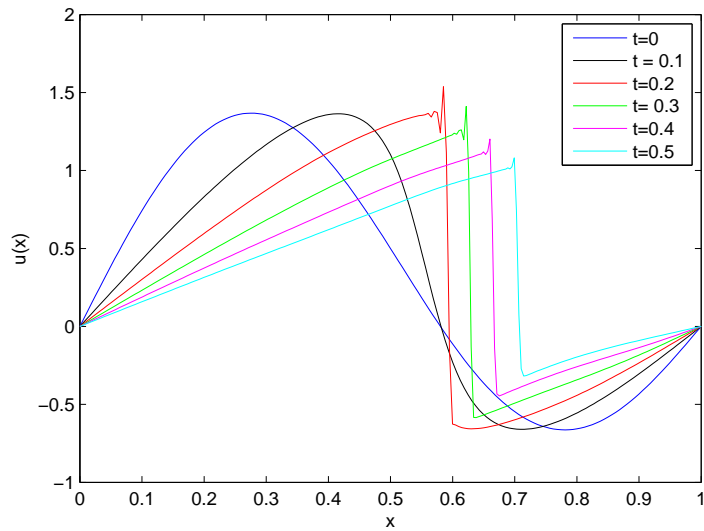
Výše popsaný postup byl aplikován ve výpočtovém prostředí MATLAB. Byly přitom využity oba přístupy k řešení. Při současném řešení modifikované MMPDE5 a Burgersovy rovnice bylo využito vnitřního implicitního řešiče v prostředí MATLAB, funkce ODE15i. Pro případ odděleného řešení bylo použito implicitní schéma pro modifikovanou MMPDE5 a explicitní schéma pro Burgersovu rovnici popsané v (2.43). Na následujících obrázcích je zobrazen průběh adaptace sítě a řešení rovnice v jednotlivých časových krocích.



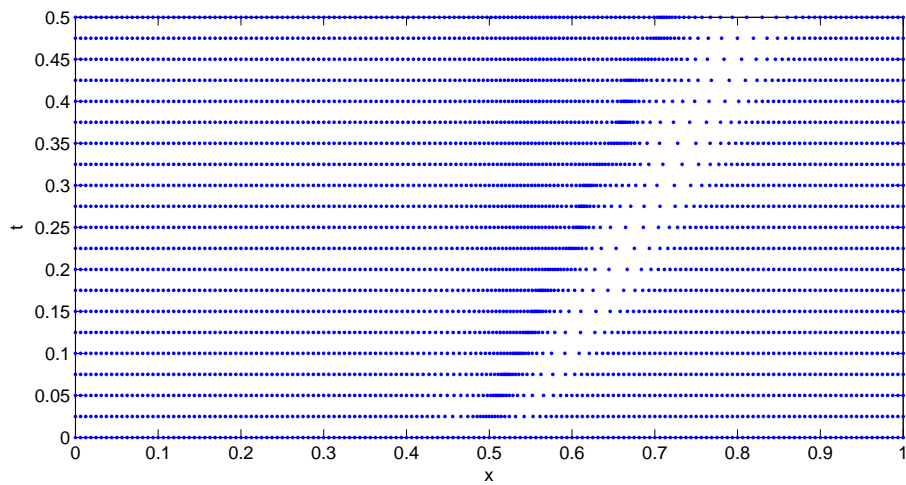
Obr. 15: Numerické řešení Burgersovy rovnice pro počáteční podmínku (2.41) průběh změn v síti



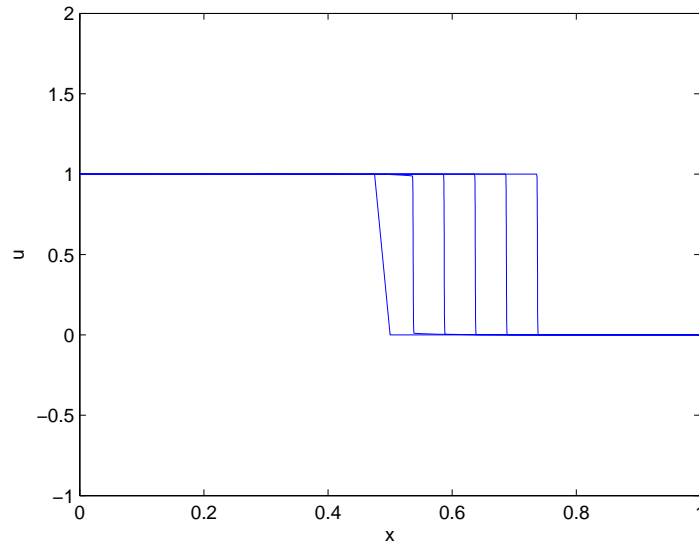
Obr. 16: Numerické řešení Burgersovy rovnice pro počáteční podmínku (2.41) pro současné řešení



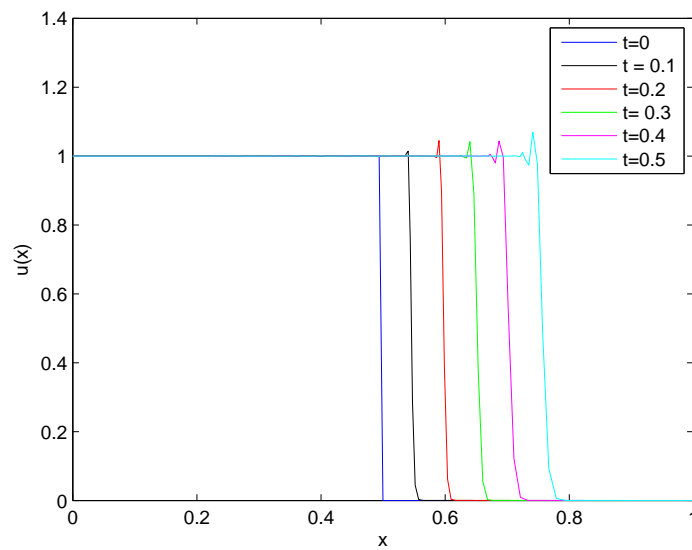
Obr. 17: Numerické řešení Burgersovy rovnice pro počáteční podmínku (2.41) pro oddělené řešení



Obr. 18: Numerické řešení Burgersovy rovnice pro počáteční podmínku (2.42) průběh změn v síti



Obr. 19: Numerické řešení Burgersovy rovnice pro počáteční podmínku (2.42) pro současné řešení



Obr. 20: Numerické řešení Burgersovy rovnice pro počáteční podmínku (2.42) pro oddělené řešení

Z výsledků je patrný rozdíl mezi současným a odděleným řešením. Při současném řešení je možné volit parametr $\tau = 10^{-2}$ čímž je zajištěna agresivnější adaptace sítě (připomeňme, že τ je jakási doba odezvy sítě na změny v řešení). V tomto případě také nenastává zpoždění popsané výše. Díky tomu se podařilo úplně odstranit oscilace v řešení.

V případě odděleného řešení je schéma stabilní řádově pro hodnoty $\tau \in (1, 100)$. Z tohoto důvodu dochází k výrazně pomalejší adaptaci sítě. Navíc vzniká výše popsané zpoždění. Z tohoto důvodu se nepodařilo úplně odstranit vzniklé oscilace, nicméně došlo alespoň k částečnému utlumení.

3 Numerické řešení proudění stlačitelných ne- vazkých a tepelně nevodivých tekutin ve 2D

Vzhledem k tomu, že práce je zaměřena na metody adaptace sítě, vystačíme si v této práci s matematickým modelem stlačitelné nevazké a tepelně nevodivé tekutiny. Tento model je popsán systémem Eulerových rovnic. Model je postačující pro zachycení rázových vln a zároveň je výpočetně méně náročný a podstatně rychleji konverguje. Systém Eulerových rovnic je zjednodušený systém Navierových-Stokesových rovnic, kde byly zanedbány vazké členy. Odvození tohoto systému je detailně popsáno například v [10] nebo [11]. Zmiňme pouze, že jak systém Navierových-Stokesových rovnic, tak systém Eulerových rovnic, vychází ze zákonů zachování hmotnosti, hybnosti (resp. momentu hybnosti) a celkové energie termodynamického systému v Eulerově popisu. Systém Eulerových rovnic lze zapsat kompaktně v integrálním tvaru jako

$$\int_{\Omega} \frac{\partial \mathbf{w}}{\partial t} d\Omega + \int_{\Omega} \frac{\partial \mathbf{F}_j^I(\mathbf{w})}{\partial x_j} d\Omega = \mathbf{0}. \quad (3.1)$$

Zde $j = 1, 2$ pro případ 2D úlohy. Proměnná t je čas a x_j je j -tá složka kartézského souřadnicového systému. Vektor \mathbf{w} nazveme vektorem konzervativních proměnných a zapíšeme jej ve tvaru

$$\mathbf{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad (3.2)$$

kde ρ je hustota, u a v jsou kartézské složky vektoru rychlosti $\mathbf{v} = (u, v)^T$ a E je celková energie termodynamického systému vztažená na jednotku objemu. Lze ji zapsat ve tvaru

$$E = \rho\epsilon + \frac{1}{2}\rho(u^2 + v^2). \quad (3.3)$$

Zde ϵ je vnitřní energie $\epsilon = c_v T$. Výraz $\frac{1}{2}\rho(u^2 + v^2)$ odpovídá měrné kinetické energii. Vektor \mathbf{F}_j^I nazveme j -tou složkou nevazkého toku. Tento vektor má pro $j = 1$ tvar

$$\mathbf{F}_1^I(\mathbf{w}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{bmatrix}, \quad (3.4)$$

respektive pro $j = 2$

$$\mathbf{F}_2^I(\mathbf{w}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{bmatrix}. \quad (3.5)$$

Tento systém Eulerových rovnic obsahuje více neznámých, než je počet rovnic. Vzhledem k tomu, že proudění stlačitelných tekutin vždy provází termodynamické změny, doplňuje se systém Eulerových rovnic o stavovou rovnici ideálního plynu ve tvaru

$$p = \rho r T, \quad (3.6)$$

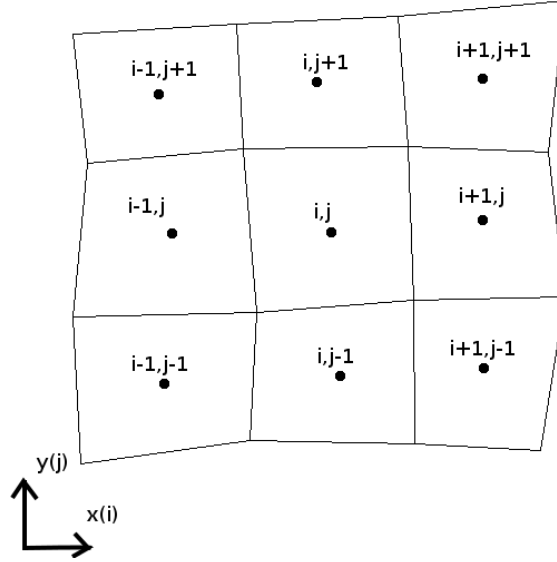
kde T je termodynamická teplota a r je specifická plynová konstanta. Stavovou rovnici lze pomocí vhodných úprav přepsat do tvaru

$$p = (\kappa - 1) \left[E - \frac{1}{2} \rho (u^2 + v^2) \right], \quad (3.7)$$

kde κ je Poissonova adiabatická konstanta.

3.1 Metoda konečných objemů

Pro prostorovou diskretizaci systému Eulerových rovnic využijeme metodu konečných objemů a odvodíme ji pro strukturovanou čtyřúhelníkovou síť. Výpočtovou oblast $\Omega \subset \mathbf{R}^2$ rozdělíme na konečný počet malých podoblastí tak, aby se tyto podoblasti navzájem nepřekrývaly a vyplňovaly celou výpočtovou oblast. Těmto podoblastem budeme říkat konečné objemy (kontrolní objemy) a v této práci budeme uvažovat jejich tvar ve formě čtyřúhelníků. V případě strukturované sítě lze zavést indexování i, j a jednotlivé objemy značit jako Ω_{ij} .



Obr. 21: Strukturovaná síť ve 2D

Řešení systému Eulerových rovnic $\mathbf{w} = \mathbf{w}(\mathbf{x}, t)$ hledáme ve středech kontrolních objemů Ω_{ij} ve formě konstantních funkcí. Tuto konstantní funkci, na kontrolním objemu Ω_{ij} , označíme jako \mathbf{w}_{ij} . Tato funkce je definována jako integrální průměr vektoru konzervativních proměnných \mathbf{w} přes kontrolní objem Ω_{ij} , tedy

$$\mathbf{w}_{ij} = \frac{1}{|\Omega_{ij}|} \int_{\Omega_{ij}} \mathbf{w}(\mathbf{x}, t) \, d\mathbf{x}. \quad (3.8)$$

Obecně představuje $|\Omega_{ij}|$ velikost kontrolního objemu Ω_{ij} , v případě řešení 2D úlohy na strukturované čtyřúhelníkové síti se jedná o obsah čtyřúhelníku. Nyní využijeme rovnice (3.1) pro jednotlivé kontrolní objemy. Převědeme integrál zahrnující nevazké toky na pravou stranu a rozepíšeme po složkách. Dostáváme

$$\int_{\Omega_{ij}} \frac{\partial \mathbf{w}}{\partial t} \, d\mathbf{x} = - \int_{\Omega_{ij}} \left[\frac{\partial \mathbf{F}_1^I(\mathbf{w})}{\partial x} + \frac{\partial \mathbf{F}_2^I(\mathbf{w})}{\partial y} \right] \, d\mathbf{x}. \quad (3.9)$$

Na pravou stranu vzniklé rovnice aplikujeme dále Greenovu větu

$$\int_{\Omega_{ij}} \frac{\partial \mathbf{w}}{\partial t} \, d\mathbf{x} = - \oint_{\partial \Omega_{ij}} [\mathbf{F}_1^I(\mathbf{w}) n_{ij}^x + \mathbf{F}_2^I(\mathbf{w}) n_{ij}^y] \, ds. \quad (3.10)$$

V této rovnici jsme zavedli hranici kontrolního objemu $\partial\Omega_{ij}$ a jednotkový vektor vnější normály k této hranici $\mathbf{n}_{ij} = (n_{ij}^x, n_{ij}^y)^T$.

Do takto upravené rovnice dosadíme integrální průměr (3.8) a dostáváme

$$\frac{d}{dt} [\mathbf{w}_{ij}(t)] | \Omega_{ij} | = - \oint_{\partial\Omega_{ij}} [\mathbf{F}_1^I(\mathbf{w}) n_{ij}^x + \mathbf{F}_2^I(\mathbf{w}) n_{ij}^y] ds. \quad (3.11)$$

Dále nahradíme křivkové integrály součtem přes jednotlivé hranice $\partial\Omega_{ij}$ kontrolního objemu Ω_{ij} . Tyto hranice označíme jako h_{ij}^m , kde pro čtyřúhelníkový kontrolní objem platí $h = 1, 2, 3, 4$. Nahrazením integrálů dostáváme

$$\frac{d}{dt} [\mathbf{w}_{ij}(t)] | \Omega_{ij} | = - \sum_{m=1}^4 [{}^m\mathbf{F}_1^I(\mathbf{w}) {}^m n_{ij}^x + {}^m\mathbf{F}_2^I(\mathbf{w}) {}^m n_{ij}^y] | h^m |. \quad (3.12)$$

Zde $\mathbf{n}_{ij}^m = ({}^m n_{ij}^x, {}^m n_{ij}^y)$ je jednotkový vektor vnější normály ke straně h_{ij}^m a $| h_{ij}^m | = \sqrt{\Delta x_m^2 + \Delta y_m^2}$ je délka strany h_{ij}^m . Vektor \mathbf{n}_{ij}^m lze zapsat jako

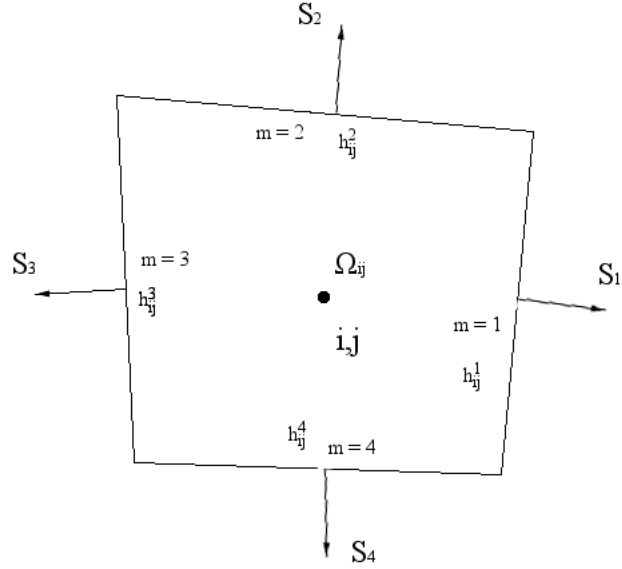
$$\mathbf{n}_{ij}^m = \left(\frac{\Delta y_m}{| h_{ij}^m |}, -\frac{\Delta x_m}{| h_{ij}^m |} \right)^T, \quad (3.13)$$

kde

- $\Delta x_m = x_{m+1} - x_m$, $\Delta y_m = y_{m+1} - y_m$ pro $m = 1, 2, 3$,
- $\Delta x_m = x_1 - x_m$, $\Delta y_m = y_1 - y_m$ pro $m = 4$.

Zavedme ještě normálový vektor \mathbf{S}_{ij}^m k jednotlivým hranám kontrolního objemu h_{ij}^m

$$\mathbf{S}_{ij}^m = \mathbf{n}_{ij}^m | h_{ij}^m | = (\Delta y_m, -\Delta x_m)^T. \quad (3.14)$$



Obr. 22: Kontrolní objem ve 2D

Nyní nahradíme nevazké toky numerickými nevazkými toky (formálně přeznačíme $\mathbf{F}_1 \rightarrow \mathbf{f}, \mathbf{F}_2 \rightarrow \mathbf{g}$) v rovnici (3.12) a dostáváme

$$\frac{d}{dt} [\mathbf{w}_{ij}(t)] |\Omega_{ij}| = - \sum_{m=1}^4 [\mathbf{f}_m^I(\mathbf{w})^m n_{ij}^x + \mathbf{g}_m^I(\mathbf{w})^m n_{ij}^y] |h^m|. \quad (3.15)$$

Nyní již záleží na volbě numerického toku. Dále je potřeba vhodně aproximovat časovou derivaci na levé straně rovnice.

3.2 Metoda časové integrace

Pro časovou diskretizaci derivace na levé straně rovnice (3.15) Využijeme Rungeovy-Kuttovy metody. Nejprve označíme pravou stranu rovnice (3.15)

jako $\mathcal{G}(\mathbf{w}(t))$, kde \mathcal{G} představuje operátor prostorové diskretizace. Touto úpravou dostáváme obyčejnou diferenciální rovnici ve tvaru

$$\frac{d}{dt} [\mathbf{w}_{ij}(t)] = \mathcal{G}(\mathbf{w}(t)). \quad (3.16)$$

Tuto rovnici řešíme pomocí explicitní Rungeovy-Kuttovy metody druhého stupně, s počáteční podmínkou ve tvaru

$$\mathbf{w}_{ij}(0) = \mathbf{w}_{ij0} = \frac{1}{|\Omega_{ij}|} \int_{\Omega_{ij}} \mathbf{w}(\mathbf{x}, 0) d\mathbf{x}. \quad (3.17)$$

Rovnici řešíme na intervalu $t = (0, \mathcal{T})$, s časovým krokem $\Delta t = t_{n+1} - t_n$. Řešení v čase t_n označíme jako \mathbf{w}_{ij}^n .

Nyní vypočteme hodnotu rovnice (3.16) a označíme ji jako

$$\mathbf{q}_1 = \mathcal{Q}(\mathbf{w}_{ij}^n). \quad (3.18)$$

Druhý krok Rungeovy-Kuttovy metody označíme jako

$$\mathbf{q}_2 = \mathcal{Q}\left(\mathbf{w}_{ij}^n + \frac{\Delta t \mathbf{q}_1}{2}\right). \quad (3.19)$$

Hodnotu \mathbf{w}_{ij}^{n+1} v čase t_{n+1} potom dostaneme jako

$$\mathbf{w}_{ij}^{n+1} = \mathbf{w}_{ij}^n + \frac{1}{2}(\mathbf{q}_1 + \mathbf{q}_2). \quad (3.20)$$

Délku časového kroku Δt určíme z nutné podmínky stability [10] ve tvaru

$$\Delta t \leq \frac{CFL}{\frac{|\lambda(\mathbf{A}_{ij})_{max}|}{\Delta x_{ij}} + \frac{|\lambda(\mathbf{B}_{ij})_{max}|}{\Delta y_{ij}}} \quad (3.21)$$

Zde CFL je konstanta volená pro explicitní schéma v intervalu $CFL \in (0, 1)$, $|\lambda(\mathbf{A}_{ij})_{max}| = |u_{ij}| + c_{ij}$ a $|\lambda(\mathbf{B}_{ij})_{max}| = |v_{ij}| + c_{ij}$ jsou maximální hodnoty vlastních čísel Jacobiových matic nevazkých toků. Proměnná c_{ij} je lokální rychlost zvuku v kontrolním objemu Ω_{ij} .

Závěrem poznamenejme, že volba Rungeovy-Kuttovy metody druhého stupně je postačující při vhodně zvolené prostorové diskretizaci. Pro jinou volbu prostorové diskretizace může být vyžadována volba vyššího řádu Rungeovy-Kuttovy metody, například řádu čtvrtého. V této práci si ovšem vystačíme s výše zmíněnou metodou.

3.3 Explicitní AUSM schéma

Pro aproximaci nevazkého numerického využijeme AUSM schéma. Odvození vychází z článků [13], [14]. Označme celkový nevazký numerický tok stěnou kontrolního objemu Ω_{ij} jako

$$\mathbf{F}^I = n_{ij}^x \cdot \mathbf{f}^I + n_{ij}^y \cdot \mathbf{g}^I. \quad (3.22)$$

Tento tok lze rozdělit na dvě různé části a to na konvektivní část $\mathbf{F}_{(c)}^I$ a tlakovou část $\mathbf{F}_{(p)}^I$, tedy

$$\mathbf{F}^I = n_{ij}^x \mathbf{f}^I + n_{ij}^y \mathbf{g}^I = \mathbf{F}_{(c)}^I + \mathbf{F}_{(p)}^I = M_n \begin{bmatrix} \rho c \\ \rho u c \\ \rho v c \\ \rho h_0 c \end{bmatrix} + p \begin{bmatrix} 0 \\ n_{ij}^x \\ n_{ij}^y \\ 0 \end{bmatrix}. \quad (3.23)$$

Zde c je lokální rychlost zvuku

$$c = \sqrt{\kappa r T}, \quad (3.24)$$

M_n je normálová hodnota machova čísla

$$M_n = \frac{V}{c}, \quad (3.25)$$

kde V je konvektivní normálová rychlost

$$V = u \cdot n_{ij}^x + v \cdot n_{ij}^y. \quad (3.26)$$

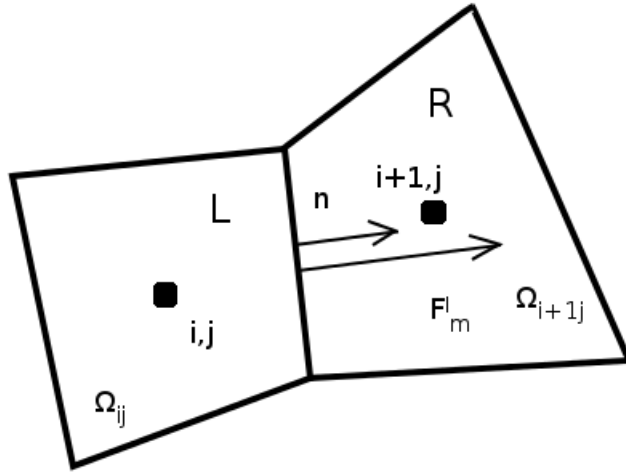
Hodnota h_0 je měrná stagnační enthalpie

$$h_0 = h + \frac{1}{2} (u^2 + v^2), \quad (3.27)$$

$$h = \epsilon + \frac{p}{\rho}, \quad (3.28)$$

kde h je měrná enthalpie.

Uvažujme dále dva kontrolní objemy ve tvaru čtyřúhelníků, které spolu sousedí, tedy mají společnou hranu h , viz následující obrázek.



Obr. 23: Sousední kontrolní objemy

Tyto dva sousedící kontrolní objemy označíme indexem L - levý (*left*) kontrolní objem a R - pravý (*right*) kontrolní objem. Stejně tak vektory konzervativních proměnných pro jednotlivé kontrolní objemy označíme jako \mathbf{w}_L , respektive \mathbf{w}_R .

Z rovnice (3.25) můžeme určit normálové Machovo číslo pro obě buňky na jejich společné stěně jako

$$M_{nL} = \frac{V_L}{c_L}, \quad (3.29)$$

$$M_{nR} = \frac{V_R}{c_R}, \quad (3.30)$$

kde V_L , respektive V_R jsou vypočteny podle rovnice (3.26). Výsledné Machovo číslo na společné stěně dvou sousedících kontrolních objemů určíme jako

$$M_{LR} = \mathcal{M}^+(M_{nL}) + \mathcal{M}^-(M_{nR}). \quad (3.31)$$

Operátor \mathcal{M} je v literatuře nazýván jako *splitting function*, volně přeloženo tedy rozkládací funkce. Existuje řada možností, jak tuto funkci zvolit. V této práci je zvolena varianta, která byla již použita v jiných pracích na KME

FAV, například v článku [15]. Zvolená funkce má tvar

$$\mathcal{M}^\pm(M) = \begin{cases} \frac{M \pm |M|}{2} & M > 1, \\ \pm \frac{(M \pm 1)^2}{4} \pm \frac{(M^2 - 1)^2}{8} & \text{jinak.} \end{cases} \quad (3.32)$$

Obdobným způsobem je definována i *splitting function* pro tlak

$$\mathcal{P}^\pm(M) = \begin{cases} \frac{M \pm |M|}{2M} & |M| > 1, \\ \frac{(M \pm 1)^2 (2 \mp M)}{4} & \text{jinak.} \end{cases} \quad (3.33)$$

Nyní lze určit tlak na společné stěně dvou sousedící kontrolních objemů jako

$$P_{LR} = \mathcal{P}^+(M_{nL}) p_L + \mathcal{P}^-(M_{nR}) p_R. \quad (3.34)$$

Poznamenejme ještě, že funkce (3.32) a (3.33) musí splňovat řadu matematických vlastností, detailně popsanych například v [15].

Se znalostí tlaků, respektive Machových čísel na společné stěně levého a pravého kontrolního objemu lze určit nevazký numerický tok například pravou stěnou h^m , kontrolního objemu Ω_{ij} jako

$$\mathbf{F}_m^I(\mathbf{w}_{ij}^L, \mathbf{w}_{i+1j}^R, \mathbf{n}_{ij}^m) = \mathbf{F}_{LR}^{I(c)} + p_{LR} \begin{bmatrix} 0 \\ m n_{ij}^x \\ m n_{ij}^y \\ 0 \end{bmatrix}. \quad (3.35)$$

Tento vztah lze přepsat do tvaru, který je možné dobře implementovat. Například opět pro pravou stěnu kontrolního objemu Ω_{ij}

$$\mathbf{F}_m^I = \frac{1}{2} M_{LR} \left(\mathbf{F}_L^{I(c)} + \mathbf{F}_R^{I(c)} \right) - \frac{1}{2} |M_{LR}| \left(\mathbf{F}_R^{I(c)} - \mathbf{F}_L^{I(c)} \right) + p_{LR} \begin{bmatrix} 0 \\ m n_{ij}^x \\ m n_{ij}^y \\ 0 \end{bmatrix}. \quad (3.36)$$

Poznamenejme ještě, že sloupcová matice $\mathbf{F}_{L(R)}^{I(c)}$ má tvar

$$\mathbf{F}_{L(R)}^{I(c)} = \begin{bmatrix} \rho c \\ \rho u c \\ \rho v c \\ \rho h_0 c \end{bmatrix}_{L(R)}. \quad (3.37)$$

Odvozené schéma AUSM, pro numerickou aproximaci nevazkých numerických toků, je prvního řádu přesnosti.

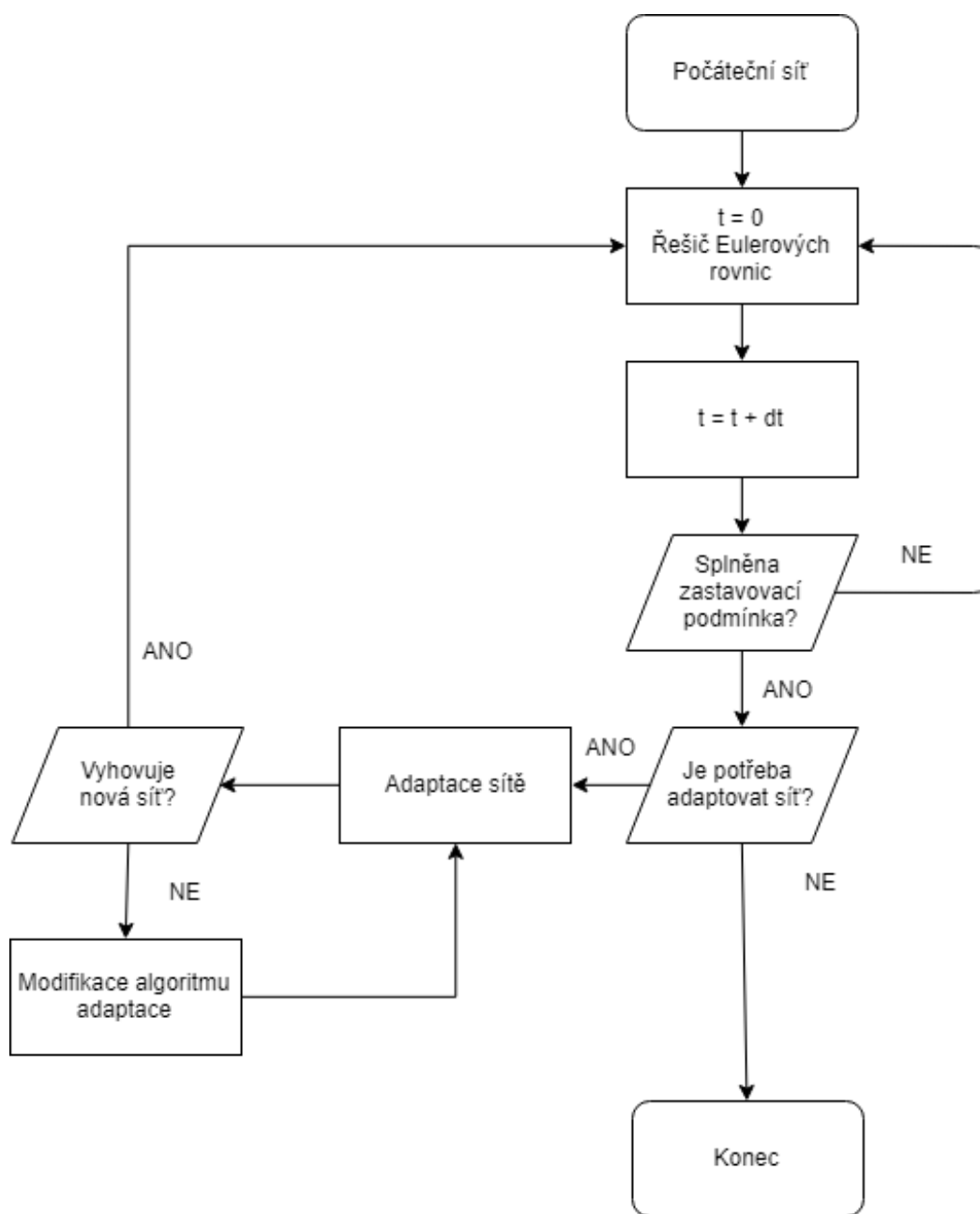
4 Adaptace výpočetní sítě ve 2D

Pro řešení úloh ve 2D byl pro adaptaci sítě zvolen přístup „h-refinement“, tedy přístup při kterém se zvyšuje počet bodů sítě, respektive počet buněk (kontrolních objemů). Tento přístup byl zvolen zejména z důvodů snadnější implementace (jak bylo zmíněno v kapitole 2, přístup „r-refinement“ je obtížně algoritmizovatelný ve dvou, respektive třech dimenzích). Cílem této kapitoly je zejména popis algoritmizace ve výpočtovém prostředí MATLAB a popis problémů, vzniklých při realizaci. Vzhledem k tomu, že tato kapitola je zaměřena zejména na metodiku a implementaci metod adaptivních sítí, vystačíme si v práci s matematickým modelem Eulerových rovnic (nevazká, stlačitelná a tepelně nevodivá tekutina). Navíc je aproximace nevazkých numerických toků jen prvního řádu přesnosti. Cílem kapitoly je tedy objasnit metodiku a implementaci a autor si uvědomuje že získané poznatky by bylo vhodné aplikovat na složitější matematické modely, respektive numerické schéma vyššího řádu přesnosti.

4.1 Implementace algoritmu pro adaptaci výpočetní sítě

Jak již bylo zmíněno a popsáno výše, pro řešení Eulerových rovnic je využito metody konečných objemů na strukturované čtyřúhelníkové síti. Tento řešič je potřeba upravit tak, aby pomocí něj bylo možné řešit problém adaptace sítě.

Vzhledem k tomu, že síť je čtyřúhelníková, je pro adaptaci sítě využito rozdělení stávajících kontrolních objemů rovnoměrně na čtyři menší kontrolní objemy. Tuto proceduru je potřeba aplikovat na takové kontrolní objemy, které byly označeny zatím neurčenou funkcí pro adaptaci. Na nově vytvořené síti je poté numericky řešen systém Eulerových rovnic, s využitím počáteční podmínky ve formě hodnot z předchozí sítě (iterace). Tento postup je opakován dokud není dosaženo požadovaného výsledku. Pro lepší ilustraci celého procesu adaptace je vhodné zobrazit proces jako vývojový diagram.



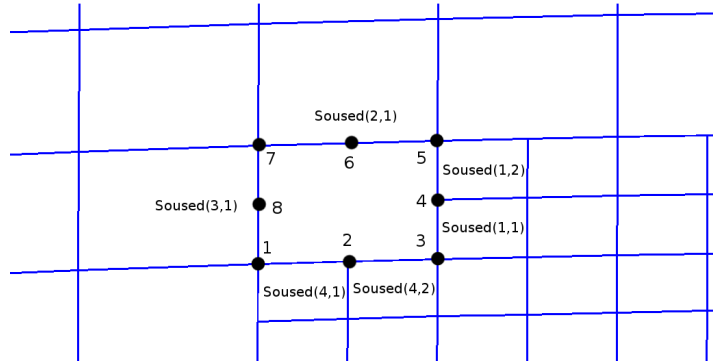
Obr. 24: Vývojový diagram průběhu výpočtu

4.1.1 Úpravy řešiče pro adaptaci sítě

Adaptování sítě vyžaduje využití nestrukturované sítě. Původní řešič, stručně popsáný v kapitole 3, je tedy potřeba upravit tak, aby podporoval adaptování sítě. Jako vhodná varianta se jeví převod řešiče, respektive sítě, na nestrukturovanou čtyřúhelníkovou síť. Strukturovanou síť, která má I buněk v i -tém směru a J buněk v j -tém směru tedy převedeme na síť která má $N = I \cdot J$ buněk, ale nelze tuto síť procházet strukturovaně.

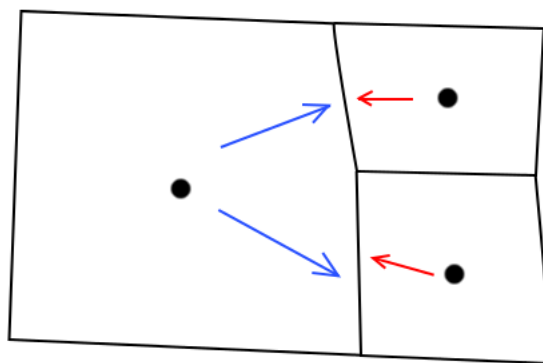
Převedením sítě do tohoto tvaru dochází ke ztrátě informace o hraničních buňkách výpočtové oblasti a také o sousedících kontrolních objemech. Z tohoto důvodu je v řešiči zavedena proměnná **sousedí**. Tato proměnná přiřadí každému kontrolnímu objemu sousední kontrolní objem, pokud má s tímto sousedním kontrolním objemem společnou hranu. Pokud řešič společnou hranu nenajde, přiřadí na volnou pozici hodnotu 0. Tato hodnota označuje, že příslušná hrana kontrolního objemu leží na hranici. Pokud řešená úloha na jedné hranici výpočetní oblasti obsahuje různé okrajové podmínky, lze na volné pozice přiřadit hodnoty -1,-2,... kde každá příslušná hodnota označuje jeden typ okrajové podmínky. Proměnná **sousedí** dále určuje, kolik kontrolních objemů leží na příslušné hraně.

Dalším problémem, který je potřeba řešit je fakt, že kontrolní objem nyní může mít více než čtyři sousedy. Protože při adaptaci nelze předem říci, kolikrát bude který kontrolní objem rozdělen, bylo potřeba omezit počet sousedů. Každý kontrolní objem může mít tedy na jedné hraně maximálně dva sousedy. S tímto omezením souvisí zvýšení počtu bodů, které definují jeden kontrolní objem na osm (u strukturované sítě stačili čtyři body).



Obr. 25: Sousedí kontrolního objemu

Pokud má kontrolní objem na jedné hraně dva (nebo více) sousedících kontrolních objemů, je potřeba dále určit závislost nevazkých numerických toků mezi jednotlivými kontrolními objemy. To znamená, že pokud má kontrolní objem na hraně dva sousedící kontrolní objemy, pak výsledný tok touto hranou je součtem toků ze sousedících buněk přes příslušící (menší) hrany.



Obr. 26: Numerické toky

4.1.2 Volba buněk určených pro adaptaci

Jedním z dalších problémů při řešení problematiky adaptace sítě je vhodná volba funkce, která vymezuje oblast pro adaptaci (v případě této práce dělení kontrolních objemů). Jako vhodná proměnná, podle které se řídí algoritmus výběru regionů pro adaptaci, se jeví hustota ρ . Algoritmus pro výběr regionů určených k adaptaci lze popsat přibližně takto.

- Pro každou hranu kontrolního objemu vypočteme relativní normu

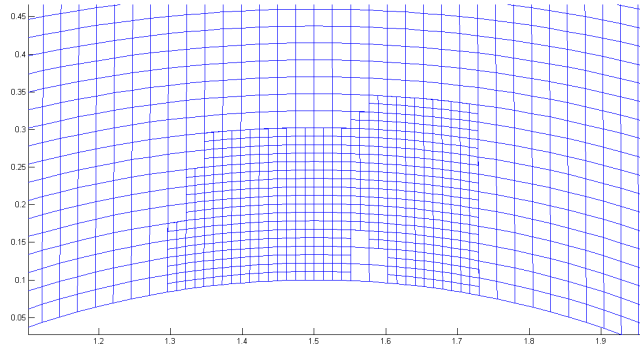
$$\varepsilon_{ik} = \left| \frac{\rho_i - \rho_k}{\rho_i} \right|, \quad (4.1)$$

kde $i = 1, 2, 3, \dots, N$ je počet buněk a k je k -tý soused kontrolního objemu.

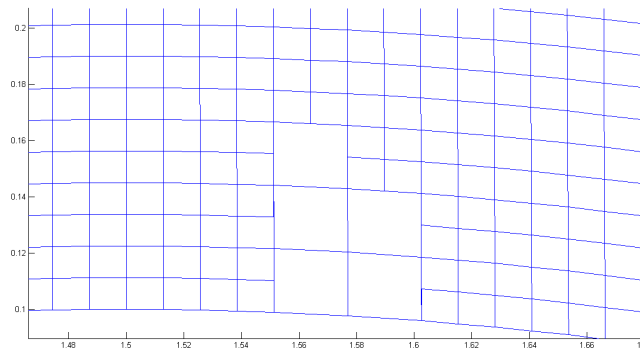
- Na každém kontrolním objemu vybereme maximální ε_{ik} .
- Pokud platí, že $\varepsilon_{ik} > C$, označíme buňku pro adaptaci. Konstanta C se může měnit s úlohou, z numerických experimentů se osvědčila hodnota řádově okolo $C = 10^{-2}$.
- Aplikujeme algoritmus pro vyplnění regionu pro adaptaci.

Algoritmus pro vyplnění regionu pro adaptaci

Pro zajištění numerické stability, při řešení systému Eulerových rovnic, je potřeba zachovat dostatečnou kvalitu sítě. Region kontrolních objemů, označených pro adaptaci, může být často nespojitý. To způsobuje skokové změny ve velikosti kontrolních objemů a tato skoková změna často vede ke ztrátě numerické stability a divergenci celého řešení. Jeden z případů, kdy nastaly problémy s numerickou stabilitou, je zobrazen na následujících obrázcích.



Obr. 27: Příklad nevhodného adaptivního regionu



Obr. 28: Příklad nevhodného adaptivního regionu - detail

Z výše popsaného důvodu je do řešiče Eulerových rovnic pro adaptivní síť implementován algoritmus, který se snaží tyto nespojitosti vyhladit, tedy vlastně doznačit kontrolní objemy uvnitř adaptivního regionu tak, aby byl tento region plně vyplněn kontrolními objemy označenými pro adaptaci.

Popsaný algoritmus hledá kontrolní objemy, které nebyly označeny pro adaptaci, ale alespoň dva jejich sousední kontrolní objemy byly pro adaptaci označeny. Tento základní algoritmus se ukázal jako dostačující a dobře funkční, pokud je proveden opakovaně, počet iterací opět záleží na zadané úloze.

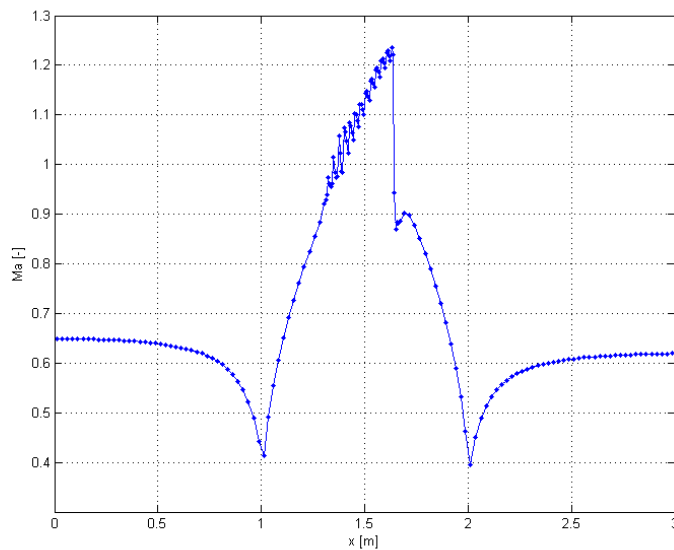
4.1.3 Tvorba adaptované sítě

Nyní přistupme k tvorbě nových kontrolních objemů. Každý kontrolní objem, označený pro adaptaci, rozdělíme rovnoměrně na 4 menší kontrolní objemy tak, aby jejich společný bod byl střed daného kontrolního objemu. Každému nově vzniklému kontrolnímu objemu přiřadíme vektor konzervativních proměnných původního, nerozděleného kontrolního objemu. Nově vzniklé adaptované kontrolní objemy poté přidáme k původním kontrolním objemům.

Po provedení tohoto algoritmu pro každý označený kontrolní objem dostáváme řadu kontrolních objemů, respektive jejich uzlů, které označují stejný kontrolní objem, respektive uzel. Z tohoto důvodu je dále nutné aplikovat algoritmus, který odstraní všechny duální kontrolní objemy, respektive uzly sítě. Tento algoritmus pouze hledá stejné hodnoty, a pokud najde dva stejné uzly/kontrolní objemy, jeden z nich odstraní.

Oprava hranice výpočtové oblasti

Při adaptování sítě na hranici výpočtové oblasti, kde je navíc tato hranice popsána křivkou, dochází ke změně geometrie vlivem chyby při adaptaci sítě. Dochází ke změně křivky na po částech lineární funkci. Ukazuje se, že tato změna má vliv na proudové veličiny u stěny a dochází k oscilacím v řešení. Z tohoto důvodu je implementován algoritmus, který opravuje nově vzniklé kontrolní objemy na hranici výpočtové oblasti. Algoritmus interpoluje nově vzniklé uzly sítě na předepsanou křivku (hranici). Tím je zachována původní geometrie.

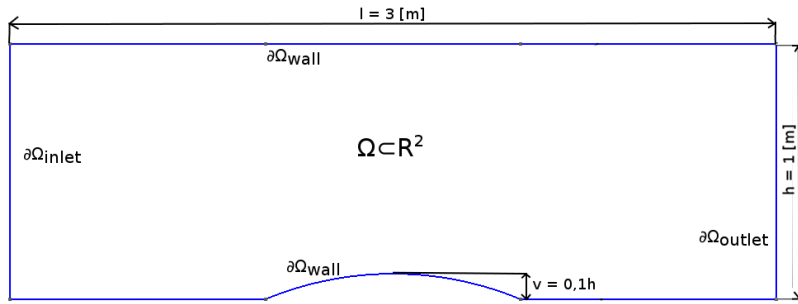


Obr. 29: Příklad oscilací na hranici výpočtové oblasti

Posledním krokem algoritmu pro adaptaci sítě je výpočet obsahů jednotlivých kontrolních objemů a normál. Tento výpočet je již ale proveden standardním způsobem jako u klasického řešení systému Eulerových rovnic tak, jak bylo popsáno v předchozí kapitole.

4.2 Numerické řešení testovacího příkladu: 2D GAMM kanál

Výše popsaný algoritmus pro adaptaci sítě byl aplikován na známou testovací úlohu, takzvaný GAMM kanál. Na následujícím obrázku je naznačena výpočtová oblast.



Obr. 30: Geometrie testovacího 2D GAMM kanálu

Hranice výpočtové oblasti $\partial\Omega_{inlet}$ představuje vstup tekutiny do výpočetní oblasti, hranice $\partial\Omega_{outlet}$ představuje výstup tekutiny z výpočetní oblasti a hranice $\partial\Omega_{wall}$ představuje pevnou nepropustnou stěnu.

Na jednotlivých stěnách předepisujeme následující okrajové podmínky.

Subsonická okrajová podmínka na vstupu

- Stagnační hustota $\rho_{s1} = 1.4637 \left[\frac{kg}{m^3} \right]$,
- stagnační tlak $p_{s1} = 137483 [Pa]$,
- úhel náběhu proudu $\beta = 0^\circ$.

Subsonická okrajová podmínka na výstupu

- Statický tlak $p_2 = 101325 [Pa]$.

Okrajová podmínka na pevné nepropustné stěně

- $\mathbf{v}^T \mathbf{n} = 0$.

Popis implementace jednotlivých okrajových podmínek je popsán v řadě publikací, například v [10]. Detailnímu popisu okrajových podmínek se podrobně věnovala také autorova bakalářská práce.

4.2.1 Adaptace 2D GAMM kanálu

Jako vstupní síť použijeme čtyřúhelníkovou nestrukturovanou síť, která má 6340 kontrolních objemů. Na této síti provedeme numerické řešení Eulerových rovnic, se zastavovací podmínkou 10 000 iterací. Tento výpočet slouží jako vstupní data pro adaptaci, proto jako zastavovací podmínku stačí předeepsat počet iterací.

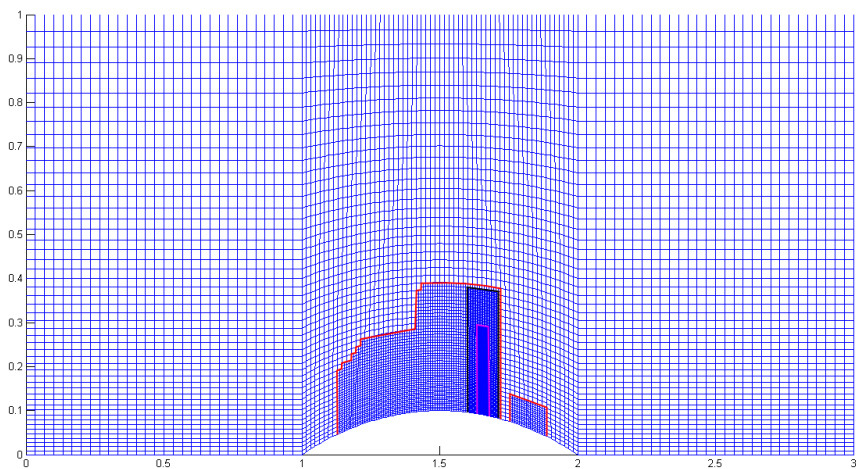
Dále budeme adaptovat síť. Kritériem konvergence je dosažení 10 000 iterací nebo splnění zastavovací podmínky

$$rez = \sqrt{\frac{\sum_{k=1}^N |\Omega_k| \frac{1}{\Delta t} (\rho_k^{n+1} - \rho_k^n)^2}{\sum_{k=1}^N |\Omega_k|}} \leq 10^{-4}. \quad (4.2)$$

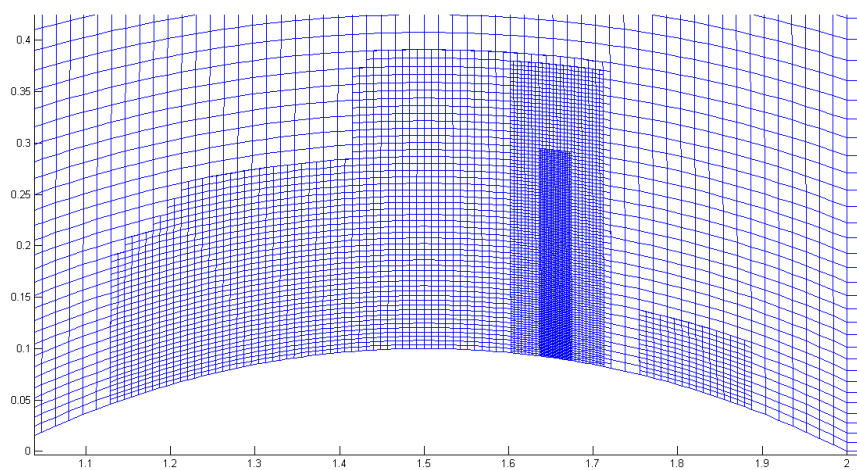
Proces adaptace je zastaven ve chvíli, kdy se podaří dostatečně přesně zachytit Zierepovu singularitu, která by měla vznikat na spodní hranici výpočtové oblasti, ve dvou třetinách kanálu, v oblasti kde dochází k prudkému poklesu Machova čísla.

4.2.2 Numerické výsledky testovacího příkladu GAMM kanálu

Výše popsaným algoritmem byly provedeny 3 adaptace. Hodnota CFL byla nastavena jako $CFL = 0.5$. Provedeny byly tři iterace adaptace sítě. Výsledná, adaptovaná síť je zobrazena na následujících obrázcích.

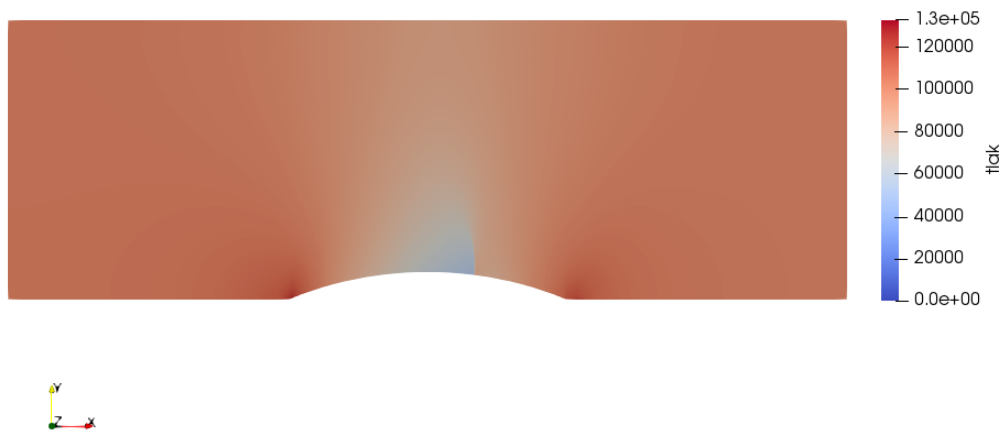


Obr. 31: Výsledná výpočetní síť po adaptaci (první adaptace červeně, druhá adaptace černě, třetí adaptace růžově)

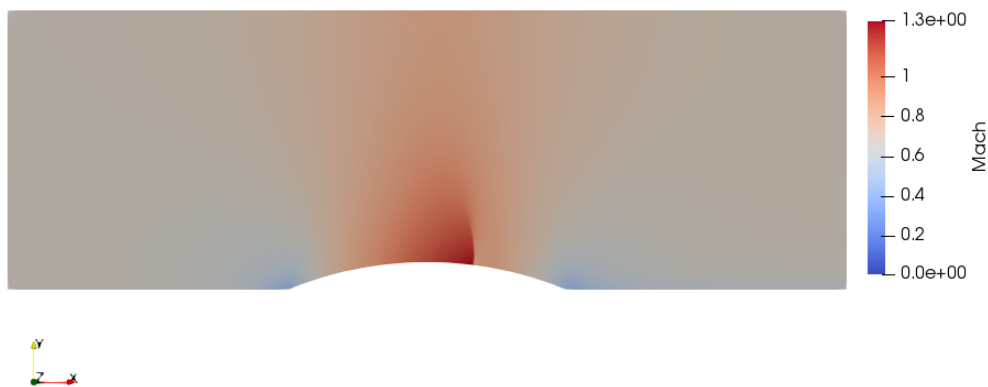


Obr. 32: Detail adaptované výpočetní sítě

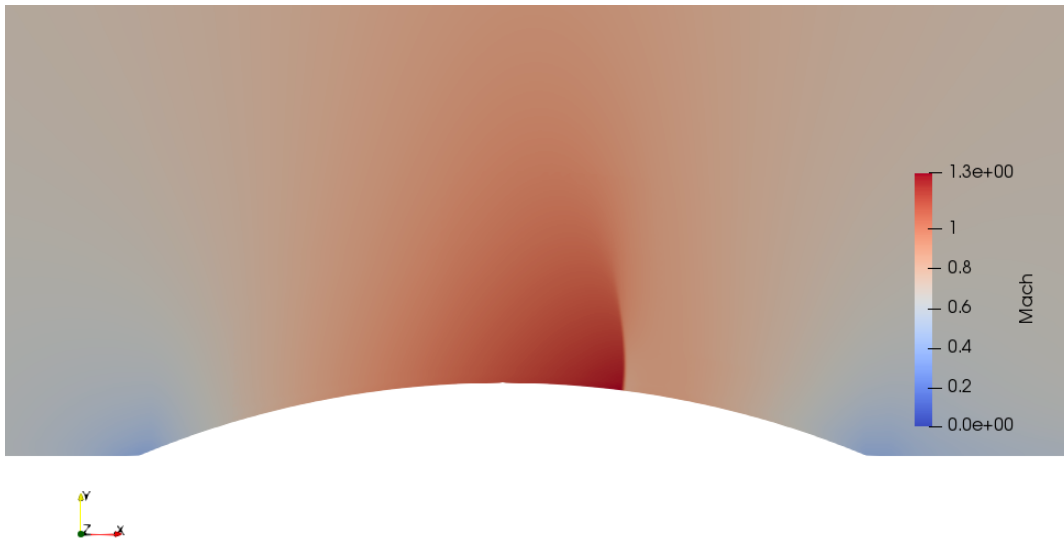
Síť se adaptovala v místě, kde se nachází rázová vlna, viz následující kontury.



Obr. 33: Kontury tlaku

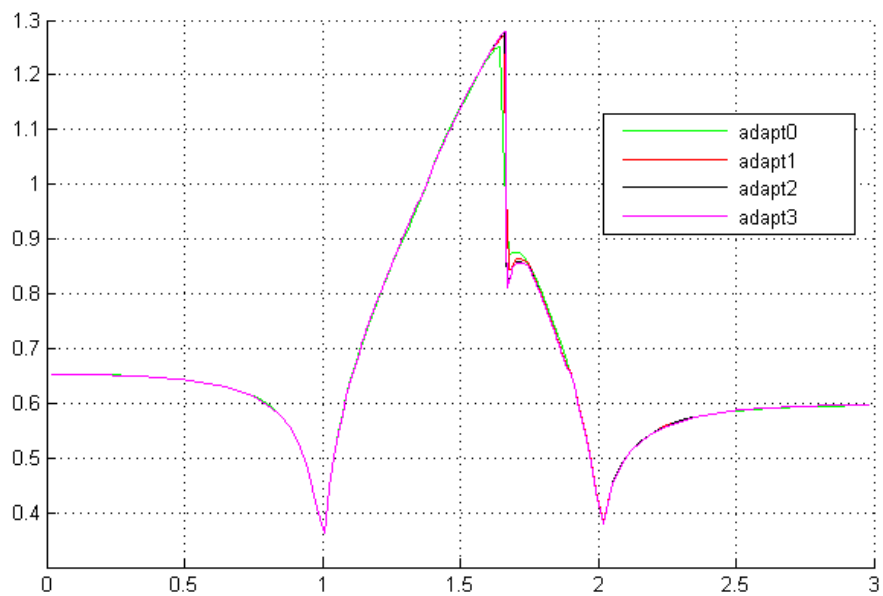


Obr. 34: Kontury Machova čísla

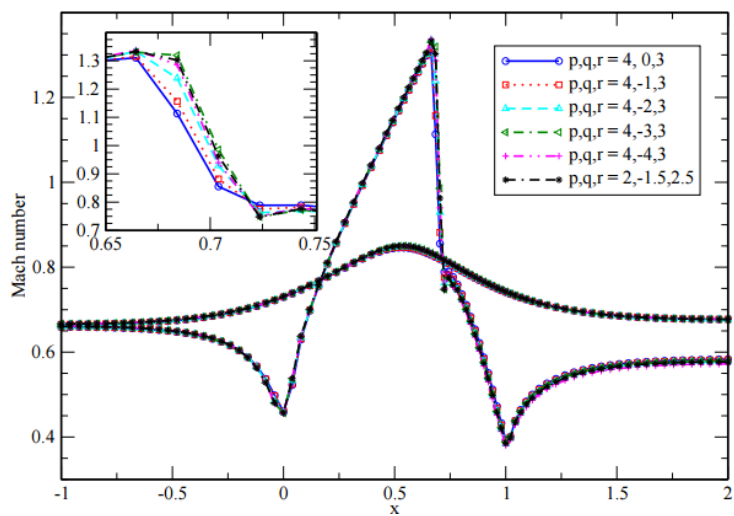


Obr. 35: Kontury Machova čísla (detail)

Dále zobrazme Machovo číslo na dolní hranici výpočtové oblasti a porovnejme ho s výsledky dosaženými v [16]

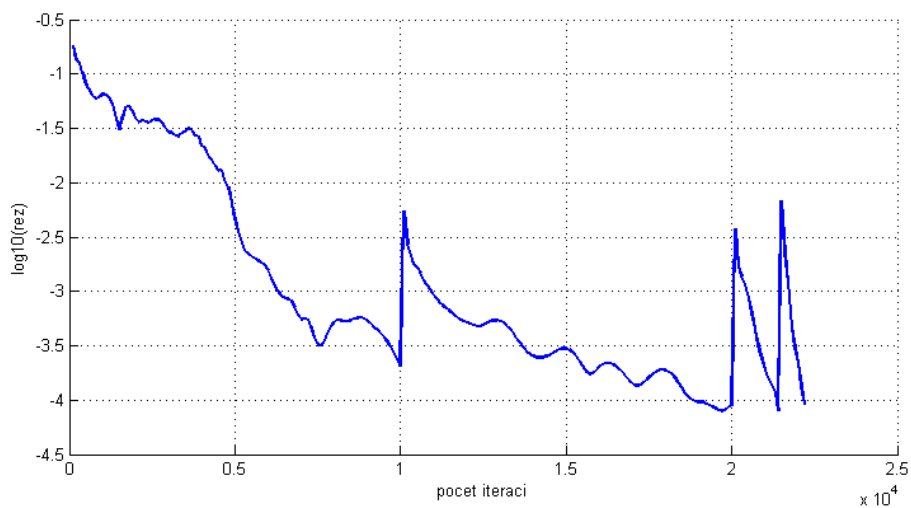


Obr. 36: Průběh Machova čísla na dolní stěně



Obr. 37: Průběh Machova čísla z literatury [16]

Jak je vidět i obrázků, podařilo se poměrně ostře zachytit Zierepovu singularitu. Machovo číslo by mělo dosahovat hodnoty 1.38, v práci dosažené výsledky dosahují hodnoty Machova čísla 1.29. Nižší hodnota je způsobena použitím AUSM schématu, které je prvního řádu přesnosti v prostoru. Pro přesnější řešení by bylo vhodné aplikovat adaptaci sítě na schéma vyššího řádu přesnosti. Na závěr této kapitoly ještě zobrazíme průběh rezidua.



Obr. 38: Průběh rezidua

Závěr

Závěrem je možné konstatovat, že cíle formulované v zadání diplomové práce byly splněny. Autor programově realizoval základní přístupy pro adaptaci výpočetní sítě typu „r-refinement“ v 1D s využitím metody konečných diferencí. Na modelovém příkladu nelineární vazké Burgersovy rovnice v 1D se podařilo snížit oscilace řešení vhodným přeskupováním uzlů sítě v průběhu výpočtu. Dále byl vyvinut program pro numerické řešení nelineárního systému Eulerových rovnic ve 2D, používající metodu konečných objemů na strukturované čtyřúhelníkové síti. Tento řešič byl výchozím pro adaptaci sítě pomocí metody typu „h-refinement“. Řešič byl upraven pro potřeby adaptaci sítě a zároveň převeden ze strukturované na nestrukturovanou čtyřúhelníkovou síť. Implementovaný adaptivní algoritmus byl použit na testovací úloze 2D GAMM kanálu. Zde se pomocí adaptace sítě podařilo poměrně ostře zachytit Zierepovu singularitu. Výsledek lze vyzdvihnout zejména z důvodu využití pouze schématu prvního řádu přesnosti pro prostorovou diskretizaci.

Numerické řešení bylo provedeno s využitím zavedených postupů pro řešení systému nelineárních Eulerových rovnic pomocí metody konečných objemů. Přínos práce lze spatřovat zejména v popisu a algoritmizaci metody pro adaptaci sítě, propojení se stávajícím řešičem a dále s popsáním problémů vznikajících při připojení adaptivního algoritmu ke klasickému řešiči.

Problematika adaptace sítě se ukazuje jako složitý problém, zejména z pohledu numerické matematiky a stability řešení. Tato práce by měla sloužit jako úvod do problematiky a vyvinutý řešič by bylo potřeba dále rozvíjet. Dalším rozšířením této práce by bylo například použití schématu pro prostorovou diskretizaci, která je vyššího řádu přesnosti. Dále by bylo velmi vhodné rozšířit stávající řešič o model Navierových-Stokesových rovnic, případně dále vylepšit algoritmus adaptace sítě tak, aby mohl tvořit například kontrolní objemy ve tvaru trojúhelníků, nebo dále zvýšit robustnost algoritmu.

Seznam použité literatury

- [1] Karlsson J.: Implementing Anisotropic Adaptive Mesh Refinement in OpenFOAM, Master's Thesis, Göteborg, Sweden, 2012.
- [2] Yi L., Premasuthan S., Jameson A.: Comparison of h- and p- Adaptations for Spectral Difference Methods Aeronautics and Astronautics Department, StanfordUniversity, USA, 2010.
- [3] Mesh adaptation. Cfd-online [online]. 2012. Dostupné z: https://www.cfd-online.com/Wiki/Mesh_adaptation.
- [4] Nissen A., Kormann K, Grandin M., Virta K.: Stable difference methods for block-oriented adaptive grids, Bergen, 2014.
- [5] Huang, W., REN Y., RUSSELL R. D.: Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, Burnaby Canada, 1994.
- [6] Gros, E.: Numerical Modelling of Two-Phase Flow with Moving Boundary Fitted Meshes, Lausanne 2018.
- [7] Aldlemy M. S., Rasani M. R., T.M.Y.S. Tuan Ya, and Ariffin A.K.: Dynamic Adaptive Mesh Refinement of Fluid Structure Interaction using Immersed Boundary Method with Two Stage Corrections, Kebangsaan Malaysia, 2018.
- [8] Huang W. Russell R.: Adaptive moving mesh methods, New York, 2011.
- [9] Thompson J. F., Warsi Z. U.A., Mastin W. C.: Numerical grid generation: foundations and applications, Elsevier North-Holland, Inc. New York, NY, USA, 1985.
- [10] Vimmr J.: Matematické modelování proudění stlačitelné tekutiny ve vnitřní aerodynamice. Disertační práce, ZČU v Plzni, 2002.
- [11] Vimmr J.: Modelování proudění tekutin s aplikacemi v biomechanice a ve vnitřní aerodynamice. Habilitační práce, ZČU v Plzni, 2008.
- [12] Liseikin V. D.: Grid Generation Methods. Springer, 2010.

- [13] Liou M. S., Steffen Ch. J.: A New Flux Splitting Scheme. *Journal of Computational Physics* 107, 23-39, 1993.
- [14] Liou M. S.: A Sequel to AUSM: AUSM+. *Journal of Computational Physics* 129: 364-382, 1996.
- [15] Hajžman M., Bublík O., Vimmr J.: On the modelling of compressible inviscid flow problems using AUSM schemes. *Applied and Computational Mechanics*, 1(2), 469- 478, 2007.
- [16] Fürst J.: Numerical solution of inviscid and viscous flows using modern schemes and quadrilateral or triangular mesh, *Proceedings of the Czech-Japanese Seminar in Applied Mathematics 2004*, Czech Technical University in Prague, 2004.