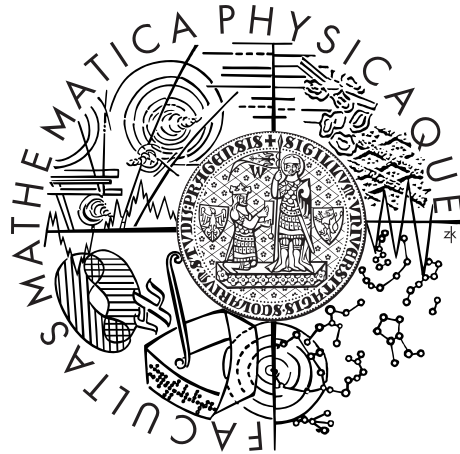


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## DISERTAČNÍ PRÁCE



Petr Kubera

# Počítačová simulace a numerická analýza problémů stlačitelného proudění

Katedra numerické matematiky

Vedoucí disertační práce: doc. RNDr. Jiří Felcman, CSc.

Studijní program: Fyzika

Studijní obor: Matematické a počítačové modelování

Praha 2011

Zde bych rád poděkoval všem svým blízkým okolo mě, kteří měli se mnou trpělivost při psaní této práce a podporovali mě. Dále bych chtěl poděkovat svému školiteli Doc. RNDr. Jiřímu Felcmanovi, CSc. za cenné rady, připomínky a v neposlední řadě čas, který mi věnoval při psaní této práce. Také bych chtěl poděkovat svému konzultantovi Doc. RNDr. Vítu Dolejšímu, Ph.D., DSc. za rady ohledně některých problémů. A dále RNDr. Karlu Findejšovi za poskytnutí implementace řešiče ADER.

Práce je částí výzkumného projektu MSM 0021620839 financovaného MŠMT a je částečně podporovaná Grantovou agenturou Univerzity Karlovy, projekty číslo 343/2005/B-MAT/MFF a 304-10/259221 MFF. Tímto jí chci poděkovat za poskytnutou podporu.

Prohlašuji, že jsem tuto disertační práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....

Podpis autora

Název práce: Počítačová simulace a numerická analýza problémů stlačitelného proudění

Autor: Mgr. Petr Kubera

Katedra: Katedra numerické matematiky

Vedoucí disertační práce: doc. RNDr. Jiří Felcman, CSc., Katedra numerické matematiky

Abstrakt: Tato práce se zabývá konstrukcí adaptivní výpočtové sítě v 1D a ve 2D v kontextu metody konečných objemů. Adaptivní strategie je aplikována na numerické řešení Eulerových rovnic, což je hyperbolický systém parciálních diferenciálních rovnic. Použitý postup je určen pro nestacionární problémy a skládá se ze tří v podstatě nezávislých kroků, jež jsou cyklicky opakovány. Těmito kroky jsou: výpočet pomocí schématu metody konečných objemů, dále pak adaptace sítě a přepočítání numerického řešení z neadaptované sítě na síť adaptovanou. Díky tomu je tento algoritmus použitelný i na jiné, nejen hyperbolické systémy. Těžiště práce spočívá v návrhu vlastní adaptační strategie, založené na anisotropní adaptivitě, která bude v každém adaptačním kroku splňovat tzv. geometrický zákon zachování. V práci je též porovnání naší navržené strategie s algoritmy typu Moving Mesh pro úlohy s pohybuující se nespojitostí.

Klíčová slova: metoda konečných objemů, adaptivní metody, geometrický zákon zachování

Title: Computer simulation and numerical analysis of compressible flow problems

Author: Mgr. Petr Kubera

Department: Department of Numerical Mathematics

Supervisor: doc. RNDr. Jiří Felcman, CSc., Department of Numerical Mathematics

Abstract: The thesis deals with the construction of an adaptive 1D and 2D mesh in the framework of the cell-centred finite volume scheme. The adaptive strategy is applied to the numerical solution of problems governed by Euler equations, which is a hyperbolic system of PDE's. The used algorithm is applicable to non-stationary problems and consists of three independent parts, which are cyclically repeated. These steps are PDE evolution, then mesh adaptation and interpolation of numerical solution from the old mesh to the newly adapted mesh. Owing to this the algorithm can be used also for other hyperbolic systems. The thesis is focused on the development of our mesh adaptation strategy, based on the anisotropic mesh adaptation, which preserves the geometric mass conservation law in each computational step. Several test problems with moving discontinuity are computed to compare our algorithm with Moving Mesh algorithms.

Keywords: finite volume method, adaptive methods, geometric mass conservation law

# Obsah

<b>Úvod</b>	<b>3</b>
<b>1 Matematický popis dynamiky tekutin</b>	<b>5</b>
1.1 Navierovy-Stokesovy rovnice . . . . .	5
1.2 Eulerovy rovnice . . . . .	7
1.2.1 Vlastnosti Eulerových rovnic . . . . .	8
1.3 Problematika počátečních a okrajových podmínek . . . . .	9
1.3.1 Okrajové podmínky pro nevazké proudění . . . . .	10
1.3.2 Počáteční podmínky . . . . .	11
1.4 Formulace problému . . . . .	11
<b>2 Metoda konečných objemů</b>	<b>12</b>
2.1 Síť konečných objemů . . . . .	12
2.1.1 Vybrané sítě a jejich konstrukce . . . . .	13
2.2 Metoda konečných objemů . . . . .	15
2.2.1 Schéma typu ADER . . . . .	16
2.2.2 Metoda konečných objemů v 1D . . . . .	17
2.3 Podmínka stability . . . . .	18
2.4 Riemannův problém . . . . .	18
2.4.1 Přibližný Riemannův řešič HLLC . . . . .	20
2.4.2 Zobecněný Riemannův problém a jeho řešení . . . . .	21
2.5 Metody Godunova typu . . . . .	23
2.5.1 Zobecněná Godunova metoda . . . . .	25
2.6 Použité numerické toky a schémata . . . . .	25
2.6.1 1D Lax-Friedrichsův numerický tok . . . . .	26
2.6.2 1D MUSCL Lax-Friedrichsův numerický tok . . . . .	26
2.6.3 1D Lax-Wendroffův numerický tok . . . . .	27
2.6.4 1D Roeův numerický tok . . . . .	28
2.6.5 Metoda ADER . . . . .	31
<b>3 Adaptace výpočetní sítě</b>	<b>33</b>
3.1 Variační metody . . . . .	33
3.1.1 Ekvidistribuce sítě . . . . .	35
3.1.2 Monitorovací funkce . . . . .	38
3.1.3 Volba monitorovacích funkcí . . . . .	43
3.1.4 Numerické řešení MMPDE v 1D . . . . .	45
3.1.5 Numerické řešení MMPDE v 2D . . . . .	48
3.1.6 Numerické experimenty v 1D . . . . .	54
3.1.7 Numerické experimenty ve 2D . . . . .	60
3.2 Anisotropní adaptivita . . . . .	76
3.2.1 Konstrukce aproximace Hessiany matice . . . . .	78
3.2.2 Minimalizace parametru kvality sítě . . . . .	80
3.3 Metody založené na minimalizaci parametru kvality vrcholu . . . . .	82
3.3.1 Metoda používající bariérovou funkci . . . . .	83
3.3.2 Metoda s kontrolou úhlů . . . . .	85

3.3.3	Metoda s opravným cyklem . . . . .	87
3.3.4	Minimalizace parametru kvality v 1D . . . . .	90
3.3.5	Numerické experimenty v 1D . . . . .	94
3.3.6	Numerické experimenty ve 2D . . . . .	102
<b>4</b>	<b>Adaptivní algoritmy</b>	<b>114</b>
4.1	Adaptivní algoritmy . . . . .	114
4.2	Transformace řešení . . . . .	116
4.2.1	Transformace řešení v 1D . . . . .	116
4.2.2	Transformace řešení ve 2D . . . . .	118
4.3	Numerické experimenty v 1D . . . . .	120
4.4	Numerické experimenty ve 2D . . . . .	135
	<b>Závěr</b>	<b>143</b>
	<b>Seznam použité literatury</b>	<b>147</b>
	<b>Seznam tabulek</b>	<b>148</b>
	<b>Seznam obrázků</b>	<b>150</b>
	<b>Použité značení a seznam zkratk</b>	<b>150</b>
	<b>Přílohová část</b>	<b>154</b>

# Úvod

Problematika proudění vazké stlačitelné tekutiny je popsána systémem parciálních diferenciálních rovnic, tzv. Navierovými-Stokesovými rovnicemi. Pro řadu problémů z oblasti dynamiky tekutin je dostačující užití jistých zjednodušujících předpokladů, jako jsou zanedbání viskozity tekutiny, vnitřních zdrojů tepla a jeho přenosu, čímž dostáváme model tzv. nevazkého stlačitelného proudění, které je popsáno Eulerovými rovnicemi. Tyto rovnice jsou, z hlediska klasifikace parciálních diferenciálních rovnic, rovnicemi hyperbolickými. Jednou z charakteristik takového systému je, že i v řešení (slabém), s hladkou počáteční podmínkou, se mohou v průběhu času objevovat nespojitosti.

Řešení systému Eulerových rovnic je důležité nejen pro speciální zjednodušené případy, ale i jako součást tzv. kombinované metody konečných prvků a objemů [12] pro řešení rovnic Navierových-Stokesových. Pro řešení systému Eulerových rovnic je nejčastěji používána metoda konečných objemů, kde je oblast, v níž uvažujeme proudění dané tekutiny, rozdělena na menší části. Tyto nazýváme konečné objemy a na těchto objemech je uvažováno řešení, které je po částech konstantní.

Klíčovým pojmem v oblasti metody konečných objemů je konstrukce tzv. numerických toků. Právě numerický tok totiž rozhoduje o přesnosti metody, přehled některých vybraných toků lze nalézt např. v knize [13]. Pro zvýšení přesnosti výpočtu je v podstatě možné postupovat dvěma směry. Prvním z nich je zvýšení řádu přesnosti numerického toku a tím i schématu metody konečných objemů, viz např. schéma ADER (Arbitrary Derivative) [46], metody ENO (Essentially Non Oscillatory), či WENO (Weighted Essentially Non Oscillatory). Druhá možnost je použití adaptivity, tj. zjemnění výpočetní sítě konečných objemů, viz např. práce [10, 40], v oblastech s nespojitostí, či s velkou hodnotou normy gradientu numerického řešení Eulerových rovnic. Ve své práci budeme používat druhého přístupu, tedy použití adaptivity, ale převážně v kombinaci se schématy metody konečných objemů vyšších řádů, konkrétně druhého řádu.

Proto, abychom sít vhodně adaptovali, je nutné určit chybu, např. *a-posteriori*, jaké jsme se při řešení dopustili, což je v případě nevazkého stlačitelného proudění velmi komplikované. Zaměříme se proto na aplikaci a rozšíření adaptivního postupu uvedeného v práci [40], tzv. Moving Mesh, který je zde použit v 1D a pro čtyřúhelníkové sítě konečných objemů ve 2D. Postup se skládá ze tří v podstatě nezávislých kroků - výpočtu pomocí metody konečných objemů, adaptace výpočetní sítě a interpolace numerického řešení mezi původní a nově adaptovanou sítí, jež jsou v průběhu výpočtu cyklicky opakovány. Důvod, proč jsme se zaměřili na tento postup, je ten, že ho lze použít i pro jiné hyperbolické systémy, kde mohou vznikat pohybující se nespojitosti, jako jsou proudění mělké vody, či problém sedimentace polydisperzních směsí.

Naším cílem bude navrhnout a na vybraných úlohách otestovat vlastní strategii adaptace sítě, jak v 1D, tak i ve 2D pro trojúhelníkové nestrukturované sítě, a porovnat je s již existujícími Moving Mesh algoritmy.

Práce je členěna následovně. V kapitole 1 je formulován základní popis matematického modelu stlačitelného proudění. V kapitole 2 je odvozena metoda konečných objemů a uvedeny některé numerické toky, které využíváme v následujících

numerických experimentech. Kapitola 3 obsahuje popis některých v současnosti užívaných postupů pro adaptaci sítě a též námi navržené původní algoritmy pro adaptaci sítě. Jak námi navržené, tak i konkurenční algoritmy, jsou zde prověřeny a porovnány ve schopnosti adaptovat výpočetní síť. V kapitole 4 jsou popsány kompletní algoritmy adaptivního výpočtu a v 1D a 2D prověřeny na vybraných úlohách s pohybujícími se nespojitostmi v řešení.

Poznamenejme, že všechny numerické experimenty byly prováděny na počítači, jehož konfigurace je uvedena v přílohové části.



# 1. Matematický popis dynamiky tekutin

Tato část práce je zaměřena na matematický popis dynamiky tekutin. Jsou zde uvedeny základní rovnice pro popis stlačitelného proudění (Navierovy-Stokesovy rovnice), podrobněji jsou zde popsány Eulerovy rovnice pro nevazké proudění, jejichž numerické řešení v kombinaci s adaptivním zjemňováním je hlavním těžištěm této práce. V souladu s [11, 13] označme časový interval, během něhož sledujeme proudění tekutiny, jako  $(0, T)$  a oblast vyplněnou tekutinou v čase  $t \in (0, T)$  jako  $\Omega_t \subset \mathbb{R}^3$ , pro oblast nezávislou na čase budeme používat  $\Omega = \Omega_t$ .

## 1.1 Navierovy-Stokesovy rovnice

Proudění stlačitelné tekutiny lze popsat pomocí následujícího systému rovnic [11, 13]:

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{v}) = 0, \quad (1.1)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \operatorname{div}(\rho \mathbf{v} \otimes \mathbf{v}) = \rho \mathbf{f} - \nabla p + \nabla(\lambda \operatorname{div} \mathbf{v}) + \operatorname{div}(2\mu \mathbb{D}(\mathbf{v})), \quad (1.2)$$

$$\begin{aligned} \frac{\partial E}{\partial t} + \operatorname{div}(E \mathbf{v}) &= \rho \mathbf{f} \cdot \mathbf{v} - \operatorname{div}(p \mathbf{v}) + \operatorname{div}(\lambda \mathbf{v} \operatorname{div} \mathbf{v}) + \\ &+ \operatorname{div}(2\mu \mathbb{D}(\mathbf{v}) \mathbf{v}) + \rho q + \operatorname{div}(k \nabla \Theta), \end{aligned} \quad (1.3)$$

$$e = e(\rho, \Theta), \quad (1.4)$$

$$p = p(\rho, \Theta), \quad (1.5)$$

kde  $\rho$  je hustota,  $\mathbf{v}$  vektor rychlosti,  $p$  tlak,  $\theta$  absolutní teplota,  $E$  celková energie,  $\mathbf{f}$  vektor hustoty vnější objemové síly,  $q$  hustota zdrojů tepla v tekutině,  $\lambda, \mu$  jsou koeficienty viskozity a  $k$  je konstanta tepelné vodivosti, viz použité značení na str. 151. Tenzor rychlosti deformace  $\mathbb{D}(\mathbf{v})$  je definován následovně

$$\mathbb{D}(\mathbf{v}) = (d_{ij})_{i,j=1}^3, \quad d_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \quad (1.6)$$

Vztahy (1.1)–(1.3) představují zákony zachování, rovnice (1.1) se nazývá *rovnice kontinuity* a je diferenciální formulací zákona zachování hmotnosti. Rovnice (1.2) je tzv. *Navierovou-Stokesovou rovnicí* a vyjadřuje diferenciální formulaci zákona zachování hybnosti. Zákon zachování energie v diferenciální formě je vyjádřen rovnicí (1.3). Zbývající rovnice (1.4), (1.5) jsou termodynamické relace, jež udávají vztahy mezi stavovými veličinami, hustotou  $\rho$ , absolutní teplotou  $\Theta$ , tlakem  $p$  a hustotou vnitřní energie  $e$ . Celý tento systém rovnic je nazýván *Navierovými-Stokesovými rovnicemi pro stlačitelné proudění*. V dalším budeme uvažovat dokonalý plyn, jež je popsán stavovou rovnicí:

$$p = R\Theta\rho, \quad (1.7)$$

kde  $R > 0$  je plynová konstanta, kterou lze vyjádřit ve tvaru:

$$R = c_p - c_v, \quad (1.8)$$

kde  $c_p$  je specifické teplo při konstantním tlaku a  $c_v$  je specifické teplo při konstantním objemu. Vztahy (1.4), resp. (1.5) budeme uvažovat v následující formě:

$$e = c_v \Theta, \quad (1.9)$$

resp.

$$p = (\gamma - 1) (E - \rho |\mathbf{v}|^2 / 2), \quad (1.10)$$

kde  $\gamma = c_p / c_v > 1$  je *Poissonova adiabatická konstanta*. Vztah mezi hustotou vnitřní energie  $e$  a celkovou energií  $E$  je dán následující relací:

$$e = \frac{E}{\rho} - \frac{1}{2} |\mathbf{v}|^2. \quad (1.11)$$

Zavedením *stavovového vektoru*  $\mathbf{w} = \mathbf{w}(x, t)$ ,  $x \in \Omega$ ,  $t \in (0, T)$ , lze tento systém rovnic ((1.1), (1.2), (1.3)) pro  $N$ -dimensionální proudění přepsat do následujícího tvaru:

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^N \frac{\partial \mathbf{f}_s(\mathbf{w})}{\partial x_s} = \mathbf{F}(\mathbf{w}) + \sum_{s=1}^N \frac{\partial \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w})}{\partial x_s}, \quad (1.12)$$

kde

$$m = N + 2,$$

$$\mathbf{w} = (w_1, w_2, \dots, w_m)^T = (\rho, \rho v_1, \dots, \rho v_N, E)^T \in \mathbb{R}^m, \quad (1.13)$$

$$\mathbf{f}_s(\mathbf{w}) = (w_{s+1}, w_2 w_{s+1} / w_1 + \delta_{1s} (\gamma - 1) (w_m - \sum_{j=2}^{m-1} w_j^2 / (2w_1)), \dots, \quad (1.14)$$

$$w_{m-1} w_{s+1} / w_1 + \delta_{m-2,s} (\gamma - 1) (w_m - \sum_{j=2}^{m-1} w_j^2 / (2w_1)),$$

$$w_{s+1} (\gamma w_m - (\gamma - 1) \sum_{j=2}^{m-1} w_j^2 / (2w_1)) / w_1)^T,$$

$$\mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}) = \sum_{j=1}^N \mathbb{K}_{ij}(\mathbf{w}) \frac{\partial w_j}{\partial x_j}. \quad (1.15)$$

Vztah (1.13) definuje stavový vektor  $\mathbf{w}$ , jehož jednotlivé složky  $w_i$  se nazývají *konzervativní proměnné*. Funkce  $\mathbf{f}_s$  v (1.14) jsou nazývány *nevazkými Eulerovými toky*. Členy  $\mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w})$  (1.15) jsou *vazké toky*. Detailní závislost vazkých toků na konzervativních proměnných a tvar matic  $\mathbb{K}_{ij}$  je uvedena např. v [13, str. 379].

Protože je nutné uvažovat pouze fyzikálně přípustné hodnoty hustoty  $\rho > 0$  a tlaku  $p > 0$ , je na definiční oblast  $D \subset \mathbb{R}^m$  funkcí  $\mathbf{f}_s$  kladen následující požadavek:

$$D = \{ \mathbf{w} \in \mathbb{R}^m; w_1 = \rho > 0, w_s = \rho v_{s-1} \in \mathbb{R} \quad s = 2, \dots, m-1, \\ w_m - \sum_{i=2}^{m-1} w_i^2 / (2w_1) = p / (\gamma - 1) > 0 \}, \quad (1.16)$$

a dále platí, že  $\mathbf{f}_s \in C^1(D)^m$ .

## 1.2 Eulerovy rovnice

*Eulerovy rovnice pro stlačitelné adiabatické proudění* popisují model nevazkého ( $\mu = \lambda = 0$ ) stlačitelného plynu, kde se zanedbává tepelná vodivost ( $k = 0$ ) a vnitřní zdroje tepla ( $q = 0$ ). Protože se uvažuje plyn, jehož hmotnost je velmi nízká, v modelu zanedbáme efekty způsobené objemovou silou ( $\mathbf{f}$ ) působící na plyn. Podobně jako v případě Navierových-Stokesových rovnic (1.12), lze zavedením stavového vektoru  $\mathbf{w}$ , viz (1.13), získat Eulerovy rovnice ve tvaru vhodném pro další analýzu:

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^N \frac{\partial \mathbf{f}_s(\mathbf{w})}{\partial x_s} = 0, \quad (1.17)$$

kde  $\mathbf{f}_s$  jsou Eulerovy nevazké toky definované v (1.14) a  $N$  je dimenze problému.

Derivace druhého členu rovnice (1.17) a užití věty o derivaci složené funkce vede na následující tvar Eulerových rovnic, což je *kvazilineární systém* parciálních diferenciálních rovnic prvního řádu:

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^N \mathbb{A}_s(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x_s} = 0, \quad (1.18)$$

kde  $\mathbb{A}_s(\mathbf{w})$  je Jakobiho maticí zobrazení  $\mathbf{f}_s$ :

$$\mathbb{A}_s(\mathbf{w}) = \frac{D\mathbf{f}_s(\mathbf{w})}{D\mathbf{w}} = \left( \frac{\partial f_{si}(\mathbf{w})}{\partial w_j} \right)_{i,j=1}^m, \quad (1.19)$$

pro  $\mathbf{w} \in D$ . Tok  $\mathcal{P}(\mathbf{w}, \mathbf{n})$  veličiny  $\mathbf{w}$  ve směru  $\mathbf{n}$  lze vyjádřit ve tvaru:

$$\mathcal{P}(\mathbf{w}, \mathbf{n}) = \sum_{s=1}^N \mathbf{f}_s(\mathbf{w}) n_s. \quad (1.20)$$

Derivování vztahu (1.20) vede na Jakobián toku  $\mathcal{P}(\mathbf{w}, \mathbf{n})$ :

$$\frac{D\mathcal{P}(\mathbf{w}, \mathbf{n})}{D\mathbf{w}} = \mathbb{P}(\mathbf{w}, \mathbf{n}) = \sum_{s=1}^N \mathbb{A}_s(\mathbf{w}) n_s. \quad (1.21)$$

Velmi důležitou charakteristikou popisující proudění je tzv. *Machovo číslo*:

$$M = \frac{|\mathbf{v}|}{a}, \quad (1.22)$$

kde  $a$  je rychlost zvuku:

$$a = \sqrt{\frac{\gamma p}{\rho}}. \quad (1.23)$$

Na základě hodnot Machova čísla se proudění rozděluje na subsonické (podzvukové), sonické a supersonické (nadzvukové), tedy:

$$M(x, t) < 1 \quad \text{nebo} \quad M(x, t) = 1 \quad \text{nebo} \quad M(x, t) > 1. \quad (1.24)$$

## 1.2.1 Vlastnosti Eulerových rovnic

V této části práce se zaměříme na popis základních vlastností Eulerových rovnic, které jsou mimo jiné využívány při konstrukci numerických schémat pro jejich řešení. Eulerovy rovnice mají následující vlastnosti:

### Homogenita

Nevazké toky  $\mathbf{f}_s$  jsou *homogenní* zobrazení prvního řádu:

$$\mathbf{f}_s(\alpha \mathbf{w}) = \alpha \mathbf{f}_s(\mathbf{w}), \quad \alpha > 0. \quad (1.25)$$

Což, v důsledku předpokladu, že  $\mathbf{f}_s \in C^1(D)^m$ , implikuje:

$$\mathbb{A}_s(\mathbf{w}) \mathbf{w} = \mathbf{f}_s(\mathbf{w}). \quad (1.26)$$

### Rotační invariantnost

Mějme libovolný vektor  $\mathbf{n} = (n_1, n_2, n_3)^T \in \mathbb{R}^3$ , tento vektor lze ve sférických souřadnicích zapsat ve tvaru:

$$\mathbf{n} = r (\cos \alpha \cos \beta, \sin \alpha \cos \beta, \sin \beta)^T, \quad (1.27)$$

kde  $r = |\mathbf{n}|$ ,  $\alpha \in [0, 2\pi)$  a  $\beta \in [-\pi/2, \pi/2]$ . Uvažujme nový souřadný systém  $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$ , který vznikne posunutím původní soustavy souřadnic  $x_1, x_2, x_3$  o  $\tilde{\sigma} \in \mathbb{R}^3$  a natočením osy  $x_1$  do směru  $\mathbf{n}$ . Transformační vztah z původní souřadné soustavy do nové souřadné soustavy je dán vztahem:

$$\begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} = \mathbb{Q}_0(\mathbf{n}) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \tilde{\sigma}, \quad (1.28)$$

kde

$$\mathbb{Q}_0(\mathbf{n}) = \begin{pmatrix} \cos \alpha \cos \beta & \sin \alpha \cos \beta & \sin \beta \\ -\sin \alpha & \cos \alpha & 0 \\ -\cos \alpha \sin \beta & -\sin \alpha \sin \beta & \cos \beta \end{pmatrix}, \quad (1.29)$$

je ortonormální matice a tedy platí  $\mathbb{Q}_0^{-1} = \mathbb{Q}_0^T$ . Stavový vektor  $\mathbf{w}$  při výše uvedené transformaci přejde na nový stavový vektor  $\mathbf{q}$ :

$$\mathbf{q} = \mathbb{Q}(\mathbf{n}) \mathbf{w}, \quad (1.30)$$

kde

$$\mathbb{Q}(\mathbf{n}) = \begin{pmatrix} 1 & & \\ & \mathbb{Q}_0 & \\ & & 1 \end{pmatrix}, \quad (1.31)$$

je ortonormální matice.

*Rotační invariantnost* se rozumí, že systém Eulerových rovnic (1.17) se při výše uvedené transformaci formálně nemění, tedy dostáváme následující systém rovnic pro  $\mathbf{q}$ :

$$\frac{\partial \mathbf{q}}{\partial t} + \sum_{s=1}^N \frac{\partial \mathbf{f}_s(\mathbf{q})}{\partial x_s} = 0. \quad (1.32)$$

Dále platí následující věty, jež jsou dokázány v [13, Theorem 3.4, Theorem 3.5]:

**Věta 1.1.** *Nechť  $\mathbf{w} \in D$  a  $\mathbf{n} \in \mathbb{R}^3$ ,  $|\mathbf{n}| = 1$ . Pak platí:*

$$\mathcal{P}(\mathbf{w}, \mathbf{n}) = \mathbb{Q}^{-1}(\mathbf{n}) \mathbf{f}_1(\mathbb{Q}(\mathbf{n})\mathbf{w}) \quad (1.33)$$

a

$$\mathbb{P}(\mathbf{w}, \mathbf{n}) = \mathbb{Q}^{-1}(\mathbf{n}) \mathbb{A}_1(\mathbb{Q}(\mathbf{n})\mathbf{w}) \mathbb{Q}(\mathbf{n}). \quad (1.34)$$

**Věta 1.2.** *Vektorová funkce  $\mathbf{w} \in C^1(Q_T)^m$  je řešením Eulerových rovnic (1.17), právě když funkce  $\mathbf{q} = \mathbf{q}(\tilde{x}, t)$  daná (1.30) splňuje transformovaný systém (1.32).*

## Hyperbolicita

**Definice 1.1.** *Systém typu (1.18) se nazývá hyperbolický v oblasti  $D \subset \mathbb{R}^m$ , jestliže všechna řešení  $\lambda_j = \lambda_j(\mathbf{w}, \mathbf{n})$ ,  $j = 1, \dots, m$ , algebraické rovnice  $m$ -tého stupně:*

$$\det \left( \lambda \mathbb{I}(\mathbf{w}) - \sum_{j=1}^N n_j \mathbb{A}_j(\mathbf{w}) \right) = 0, \quad (1.35)$$

kde  $\mathbb{I}$  je jednotková matice, jsou reálná pro libovolné  $\mathbf{n} = (n_1, \dots, n_N)^T \in \mathbb{R}^N$  a  $\mathbf{w} \in D$ .

V případě, že vlastní čísla  $\lambda_j$  jsou navíc jednoduchá, řekneme, že systém (1.18) je ostře hyperbolický.

Systém (1.18) je nazýván (ostře) diagonálně hyperbolický v případě, že matice  $\mathbb{P}$  (viz (1.21)) je navíc diagonalizovatelná, tedy existuje taková nesingulární matice  $\mathbb{T}$ , že platí:

$$\mathbb{T}^{-1} \mathbb{P} \mathbb{T} = \Lambda = \Lambda(\mathbf{w}, \mathbf{n}) = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix}. \quad (1.36)$$

Systém Eulerových rovnic (1.17) je diagonálně hyperbolický ve smyslu předchozí definice (1.1). Vlastní čísla  $\lambda$  pro 3D problém jsou následující:

$$\begin{aligned} \lambda_1(\mathbf{w}, \mathbf{n}) &= \mathbf{v} \cdot \mathbf{n} - a|\mathbf{n}|, \\ \lambda_2(\mathbf{w}, \mathbf{n}) &= \lambda_3(\mathbf{w}, \mathbf{n}) = \lambda_4(\mathbf{w}, \mathbf{n}) = \mathbf{v} \cdot \mathbf{n}, \\ \lambda_5(\mathbf{w}, \mathbf{n}) &= \mathbf{v} \cdot \mathbf{n} + a|\mathbf{n}|, \end{aligned} \quad (1.37)$$

kde  $a$ , které je dáno vztahem (1.23), je rychlost zvuku. Pro 2D problém lze získat následující vlastní čísla  $\lambda$  matice  $\mathbb{P}$ :

$$\begin{aligned} \lambda_1(\mathbf{w}, \mathbf{n}) &= \mathbf{v} \cdot \mathbf{n} - a|\mathbf{n}|, \\ \lambda_2(\mathbf{w}, \mathbf{n}) &= \lambda_3(\mathbf{w}, \mathbf{n}) = \mathbf{v} \cdot \mathbf{n}, \\ \lambda_4(\mathbf{w}, \mathbf{n}) &= \mathbf{v} \cdot \mathbf{n} + a|\mathbf{n}|. \end{aligned} \quad (1.38)$$

## 1.3 Problematika počátečních a okrajových podmínek

V této části práce se zaměříme pouze na základní popis problematiky okrajových a počátečních podmínek. Protože se zabýváme numerickým řešením problémů

nevazkého proudění, bude následující text věnován problematice okrajových a počátečních podmínek pro nevazké proudění.

Jako  $\Gamma_W \subset \partial\Omega$  označíme pevnou nepropustnou stěnu, jako  $\Gamma_I \subset \partial\Omega$  resp.  $\Gamma_O \subset \partial\Omega$  označíme tu část hranice  $\partial\Omega$ , kterou proudící tekutina do oblasti vstupuje<sup>1</sup>, resp. z ní vystupuje<sup>2</sup>:

$$\Gamma_I(t) = \{x \in \partial\Omega; \mathbf{v}(x, t) \cdot \mathbf{n}(x) < 0\}, \quad (1.39)$$

$$\Gamma_O(t) = \{x \in \partial\Omega; \mathbf{v}(x, t) \cdot \mathbf{n}(x) > 0\}, \quad (1.40)$$

kde  $\mathbf{n}$  je jednotková vnější normála k  $\partial\Omega$ .

### 1.3.1 Okrajové podmínky pro nevazké proudění

V případě nevazkého proudění na oblasti  $\Omega$  se obecně uvažuje Dirichletova podmínka pro stavový vektor  $\mathbf{w}$  na hranici  $\partial\Omega$

$$\mathbf{w}|_{\partial\Omega} = \mathbf{w}_D, \quad (1.41)$$

kde  $\mathbf{w}_D$  je dáno. Vzhledem k charakteru rovnic není možné zvolit libovolnou hodnotu  $\mathbf{w}_D$ , ale takovou, která respektuje jejich charakter, tj. některé z hodnot stavového vektoru se získávají extrapolací.

To, které veličiny lze při numerickém řešení na hranici  $\partial\Omega$  předepsat a které se musí extrapolovat, určíme podle počtu kladných a záporných vlastních čísel  $\lambda(\mathbf{w}, \mathbf{n})$  Jakobiho matice  $\mathbb{P}(\mathbf{w}, \mathbf{n})$  (viz (1.21)). Na hranici se předepisuje tolik veličin, kolik je záporných vlastních čísel  $\lambda(\mathbf{w}, \mathbf{n})$ , a ostatní veličiny se získají extrapolací. Poznamenejme, že v dalším textu uvažujeme okrajové podmínky pro případ  $N = 3$ , pro  $N = 2$  je situace analogická.

Ze vztahů (1.37), které určují vlastní čísla Jakobiho matice  $\mathbb{P}(\mathbf{w}, \mathbf{n})$  pro 3D problém, a z požadavku  $(\mathbf{v} \cdot \mathbf{n}) = 0$ , který platí pro pevnou část hranice  $\Gamma_W$ , vyplývá, že  $\lambda_1 < 0$ ,  $\lambda_2 = \lambda_3 = \lambda_4 = 0$ ,  $\lambda_5 > 0$ . Dle výše uvedeného předepisujeme pouze jednu veličinu, ostatní veličiny, jako jsou tlak  $p$ , hustota  $\rho$  a tečné složky rychlosti  $v_{t_1}$  a  $v_{t_2}$ , se extrapolují.

V případě vstupu a výstupu je třeba ještě rozlišit, zda se jedná o proudění subsonické, sonické či supersonické. Pro vstupní část hranice  $\Gamma_I$  v případě supersonického proudění  $(-\mathbf{v} \cdot \mathbf{n} > a)$  jsou vlastní čísla následující  $\lambda_1 < 0$ ,  $\lambda_2 = \lambda_3 = \lambda_4 < 0$ ,  $\lambda_5 < 0$ , a tedy na hranici se předepíše následující veličiny:

$$\rho|_{\Gamma_I} = \rho_D, \quad \mathbf{v}|_{\Gamma_I} = \mathbf{v}_D, \quad p|_{\Gamma_I} = p_D. \quad (1.42)$$

V případě proudění subsonického, či sonického  $(-\mathbf{v} \cdot \mathbf{n} \leq a)$  jsou vlastní čísla  $\lambda_1 < 0$ ,  $\lambda_2 = \lambda_3 = \lambda_4 < 0$ ,  $\lambda_5 \geq 0$ , a tedy na vstupní části hranice  $\Gamma_I$  se předepisují tyto veličiny:

$$\rho|_{\Gamma_I} = \rho_D, \quad \mathbf{v}|_{\Gamma_I} = \mathbf{v}_D \quad (1.43)$$

a hodnota tlaku  $p$  na hranici se získá extrapolací

Pro výstupní část hranice  $\Gamma_O$  se v případě supersonického(sonického) proudění  $(\mathbf{v} \cdot \mathbf{n} \geq a)$  nepředepisuje nic ( $\lambda_1 > 0$ ,  $\lambda_2 = \lambda_3 = \lambda_4 > 0$ ,  $\lambda_5 > 0$ ), veličiny  $\rho$ ,  $\mathbf{v}$ ,

---

<sup>1</sup>inlet

<sup>2</sup>outlet

$p$  se extrapolují. V případě subsonického proudění ( $\mathbf{v} \cdot \mathbf{n} < a$ ) jsou vlastní čísla  $\lambda_1 < 0$ ,  $\lambda_2 = \lambda_3 = \lambda_4 > 0$ ,  $\lambda_5 > 0$ , a tedy se pouze předepisuje hodnota tlaku na výstupní hranici  $\Gamma_O$ :

$$p|_{\Gamma_O} = p_D, \quad (1.44)$$

ostatní hodnoty  $\rho$ ,  $\mathbf{v}$  se získají extrapolací.

### 1.3.2 Počáteční podmínky

Počáteční podmínky (1.45) udávají počáteční rozložení fyzikálních veličin hustoty  $\rho$ , vektoru rychlosti  $\mathbf{v}$  a teploty  $\Theta$  na oblasti  $\Omega$  v čase  $t = 0$ .

$$\rho(x, 0) = \rho^0(x), \quad \mathbf{v}(x, 0) = \mathbf{v}^0(x), \quad \Theta(x, 0) = \Theta^0(x), \quad x \in \Omega. \quad (1.45)$$

Tyto podmínky lze při použití stavového vektoru  $\mathbf{w}$  přepsat následovně:

$$\mathbf{w}(x, 0) = \mathbf{w}^0(x), \quad x \in \Omega. \quad (1.46)$$

## 1.4 Formulace problému

Cílem práce je numericky řešit soustavu Eulerových rovnic:

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^N \frac{\partial \mathbf{f}_s(\mathbf{w})}{\partial x_s} = 0 \quad \text{v} \quad Q_T = \Omega \times (0, T), \quad (1.47)$$

za počáteční podmínky

$$\mathbf{w}(x, 0) = \mathbf{w}^0(x), \quad x \in \Omega \quad (1.48)$$

a okrajových podmínek

$$B(\mathbf{w}(x, t)) = 0 \quad \text{pro} \quad (x, t) \in \partial\Omega \times (0, T), \quad (1.49)$$

kde  $B(\mathbf{w}(x, t))$  je vhodný operátor popisující okrajové podmínky.

## 2. Metoda konečných objemů

V této části práce budeme formulovat popis *metody konečných objemů* pro nestrukturované sítě<sup>1</sup>, která se používá pro řešení Eulerových rovnic (1.17.). Je zde uvedeno její odvození, podmínka stability a některé vybrané numerické toky, jež jsou použity v numerických experimentech v následujících kapitolách. Formulace metody konečných objemů a pojmu sítě konečných objemů je provedena co nejobecněji, tedy pro  $N = 3$ . Konkrétní numerické toky uvádíme pouze takové, které jsme použili v numerických experimentech, tedy pro  $N = 1, 2$ .

### 2.1 Síť konečných objemů

Buď  $\Omega \subset R^N$  oblast vyplněná tekutinou,  $\Omega_h$  pak označuje polygonální, resp. polyedrickou aproximaci oblasti  $\Omega$  pro  $N = 2$ , resp.  $N = 3$ . Hranice této aproximované oblasti je pak označena  $\partial\Omega_h$  a je tvořena konečným počtem po částech lineárních křivek, resp. ploch pro  $N = 2$ , resp.  $N = 3$ . Jako *síť konečných objemů* je nazývána množina:

$$\mathcal{D}_h = \{D_i\}_{i \in J}, \quad (2.1)$$

kde  $J \subset Z^+ = \{0, 1, \dots\}$  je indexovou množinou,  $h > 0$  a  $D_i$  jsou uzavřené polygony, resp. polyedry pro  $N = 2$ , resp.  $N = 3$ , které tvoří  $\Omega_h$  a mají vzájemně disjunktní vnitřky<sup>2</sup>. Prvky  $D_i$  jsou nazývány *konečné objemy* a platí pro ně:

$$\bar{\Omega}_h = \bigcup_{i \in J} D_i, \quad (2.2)$$

kde hodnota  $h$  značí:

$$h = \sup_{i \in J} \text{diam}(D_i). \quad (2.3)$$

V dalším bude pomocí  $|D_i|$  značena  $N$ -dimenzionální míra konečného objemu  $D_i \in \mathcal{D}_h$  a pomocí  $|\partial D_i|$   $N - 1$  dimenzionální míra hranice objemu  $\partial D_i$ .

Libovolné dva konečné objemy  $D_i$  a  $D_j$  jsou buď disjunktní, nebo je jejich průnikem část hranice  $\partial D_i$  a  $\partial D_j$ . Dva konečné objemy, jejichž průnik  $\partial D_i \cap \partial D_j$  obsahuje buď úsečku, nebo část roviny se nazývají *sousední konečné objemy*. Množina indexů sousedních objemů příslušejících k objemu  $D_i$  je označena  $s(i)$ , tedy:

$$s(i) = \{j \in J; j \neq i, D_j \text{ je sousem k } D_i\}. \quad (2.4)$$

Společná hranice dvou sousedních objemů  $D_i$  a  $D_j$  se označuje jako  $\Gamma_{ij}$  a platí pro ni:

$$\Gamma_{ij} = \Gamma_{ji}. \quad (2.5)$$

Obecně je možné uvažovat takové společné hranice sousedních objemů, které jsou tvořeny konečným počtem  $\beta_{ij}$  úseček, resp. rovin pro  $N = 2$ , resp.  $N = 3$ :

$$\Gamma_{ij} = \bigcup_{\alpha=1}^{\beta_{ij}} \Gamma_{ij}^\alpha. \quad (2.6)$$

<sup>1</sup>Pojem strukturovanosti či nestrukturovanosti sítě je diskutován dále v této kapitole

<sup>2</sup>Protože se v pozdějších kapitolách zabýváme i 1D výpočty, je důležité zmínit, že v případě jedné dimenze je samozřejmě nutné terminologii o "řád snížit".



Podobně jako výše označme  $N - 1$  dimenzionální míru  $\Gamma_{ij}^\alpha$  pomocí  $|\Gamma_{ij}^\alpha|$ .

Pro části průniku hranice  $\partial\Omega_h$  s hranicí objemu  $\partial D_i$ ,  $i \in J$ , které jsou tvořeny buď úsečkou nebo částí roviny, budeme používat označení  $S_j$ , kde  $j$  je prvkem indexové množiny  $J_B \subset Z^- = \{-1, -2, \dots\}$ . Nyní lze definovat množinu indexů hraničních segmentů  $\gamma(i)$  pro objem  $D_i$  následovně:

$$\gamma(i) = \begin{cases} \{j \in J_B; S_j \subset \partial\Omega_h \cap \partial D_i\} & D_i \text{ je přilehlý k } \partial\Omega_h \\ \emptyset & D_i \text{ není přilehlý k } \partial\Omega_h \end{cases}. \quad (2.7)$$

Při použití označení  $\Gamma_{ij} = \Gamma_{ij}^1 = S_j$  a  $\beta_{ij} = 1$  pro  $j \in \gamma(i)$  lze zavést množinu  $S(i)$  jako sjednocení množin  $\gamma(i)$  a  $s(i)$ , tedy:

$$S(i) = s(i) \cup \gamma(i). \quad (2.8)$$

Pro hranci objemu  $\partial D_i$  pak platí:

$$\partial D_i = \bigcup_{j \in S(i)} \bigcup_{\alpha=1}^{\beta_{ij}} \Gamma_{ij}^\alpha \quad (2.9)$$

a také

$$|\partial D_i| = \sum_{j \in S(i)} \sum_{\alpha=1}^{\beta_{ij}} |\Gamma_{ij}^\alpha|. \quad (2.10)$$

### 2.1.1 Vybrané sítě a jejich konstrukce

Nyní se zaměříme na některé nejčastěji používané konstrukce sítí ve 2D a jejich klasifikaci. Poznamenejme, že v literatuře se vyskytuje nejednoznačnost v terminologii, např. síť konformní a síť regulární [13, str. 187, str. 195 a str. 444].

#### Trojúhelníková síť

Jednou z nejjednodušších sítí je trojúhelníková síť, kde jednotlivé objemy  $D_i$  jsou uzavřené trojúhelníky a výsledná polygonální aproximace  $\Omega_h$  oblasti  $\Omega$  je triangulací. Velmi často je na trojúhelníkovou síť kladen požadavek, aby síť byla *konformní*. Tedy aby pro  $D_i, D_j \in \mathcal{D}_h$  kde  $D_i \neq D_j$  platila právě jedna z podmínek:

$$D_i \cap D_j \text{ je } \begin{cases} \text{buď } \emptyset \\ \text{nebo je společný vrchol } D_i \text{ a } D_j \\ \text{nebo je společnou hranou } D_i \text{ a } D_j, \end{cases} \quad (2.11)$$

a tedy pro konformní trojúhelníkovou síť platí, že počet segmentů hranic společné hranice objemů  $D_i$  a  $D_j$   $\beta_{ij} = 1$ . V dalším textu budeme trojúhelníkovou konečně objemovou síť značit  $\mathcal{T}_h$ .

#### Čtyřúhelníková síť

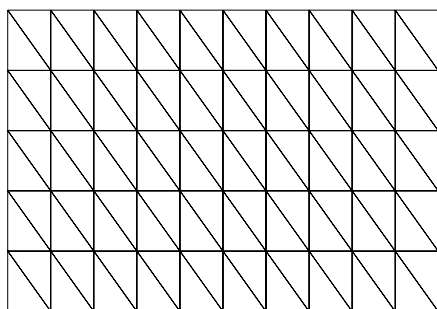
Čtyřúhelníková síť je taková síť, kde jednotlivé objemy  $D_i$  tvořící polygonální aproximaci  $\Omega_h$  jsou čtyřúhelníky a je splněna podmínka konformnosti (2.11).

## Duální konečně objemová síť nad triangulací

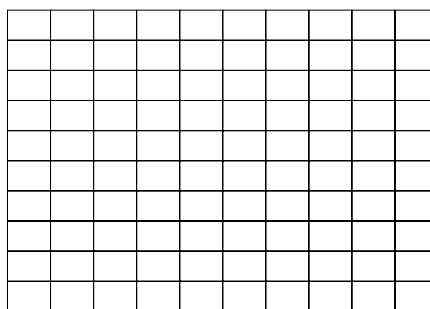
Duální konečně objemovou síť lze vytvořit za předpokladu, že již nějaká výše zmíněná konformní triangulace  $\mathcal{T}_h$  existuje. Jako duální objem  $D_i$  příslušný vrcholu  $P_i$  trojúhelníkové sítě  $\mathcal{T}_h$  budeme nazývat uzavřený polygon zkonstruovaný následovně. Spojíme těžiště každého trojúhelníku  $K \in \mathcal{T}_h$ , který obsahuje vrchol  $P_i$ , se středem každé hrany trojúhelníku  $K$ , která obsahuje vrchol  $P_i$ . Pro vrcholy  $P_i$ , jež leží na hranici  $\partial\Omega_h$ , se zkonstruovaná část polygonu uzavře pomocí odpovídajícího segmentu hranice. Výsledné polygony pak tvoří jednotlivé duální konečné objemy  $D_i$ . Pro takto zkonstruované objemy  $D_i$  a  $D_j$ , kde  $j \in s(i)$ , platí, že  $\beta_{ij} \leq 2$ .

## Barycentrická konečně objemová síť nad triangulací

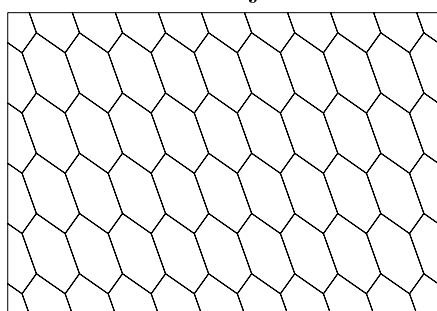
Předpokládejme opět, že máme zkonstruovanou konformní triangulaci  $\mathcal{T}_h$ . Barycentrická síť k této triangulaci se zkonstruuje tak, že pro každou nehraniční hranu původní triangulace  $\mathcal{T}_h$  spojíme její krajní body s těžišti přilehlých trojúhelníkových objemů, tj. objemů, kterým je tato hrana společná. Tímto vznikne množina hran formující barycentrický objem  $D_i$ . V případě hraničních hran, tj. hran, jejichž všechny body leží na hranici  $\partial\Omega_h$ , se postupuje obdobně. Navíc se tyto hrany přidávají do množiny hran formující příslušný barycentrický konečný objem. Pro takto zkonstruované objemy  $D_i$  a  $D_j$ , kde  $j \in s(i)$ , platí, že  $\beta_{ij} = 1$ .



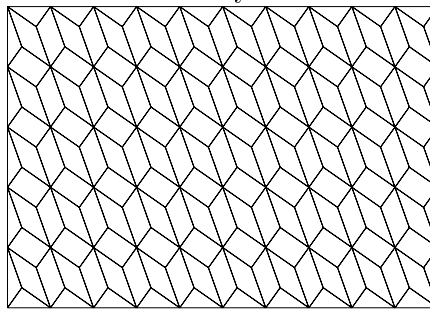
Obrázek 2.1: Trojúhelníková síť



Obrázek 2.2: Čtyřúhelníková síť



Obrázek 2.3: Duální síť



Obrázek 2.4: Barycentrická síť

## Další charakteristiky konečně objemových sítí

Dle [13, str. 195] je možné uvést další charakteristiky konečně objemových sítí:

- *Strukturované* sítě jsou takové sítě, u kterých se v každém z vrcholů sítě stýká stejný počet objemů.

- *Nestrukturované* sítě jsou sítě, kde v každém z uzlů sítě se může stýkat různý počet objemů.
- *Uniformní* síť je taková, kde jsou všechny konečné objemy stejného typu.
- *Neuniformní* síť je taková, kde nejsou všechny konečné objemy stejného typu.
- *Izotropní* sítě jsou takové sítě, kde délková měřítka konečných objemů jsou v podstatě všude v síti stejná.
- *Anizotropní* sítě jsou takové sítě, kde se délková měřítka konečných objemů sítě výrazně liší v závislosti na směru, či poloze.

## 2.2 Metoda konečných objemů

V dalším předpokládejme, že  $\mathbf{w} : \bar{\Omega} \times [0, T] \rightarrow R^m$  je klasickým řešením problému (1.4),  $\mathcal{D}_h = \{D_i\}_{i \in J}$  je konečně objemová síť a  $0 < t_0 < t_1 < \dots$  je dělení časového intervalu  $[0, T]$ . Označme  $\tau_k = t_{k+1} - t_k$  časový krok mezi  $t_k$  a  $t_{k+1}$ .

Integrací rovnice (1.17) přes množinu  $D_i \times (t_k, t_{k+1})$  a užitím Greenovy věty dostáváme:

$$\int_{D_i} (\mathbf{w}(x, t_{k+1}) - \mathbf{w}(x, t_k)) dx + \int_{t_k}^{t_{k+1}} \left( \int_{\partial D_i} \sum_{s=1}^N \mathbf{f}_s(\mathbf{w}) n_s dS \right) dt = 0,$$

vezmeme-li v úvahu (2.9), dostáváme:

$$\int_{D_i} (\mathbf{w}(x, t_{k+1}) - \mathbf{w}(x, t_k)) dx + \int_{t_k}^{t_{k+1}} \left( \sum_{j \in S(i)} \sum_{\alpha=1}^{\beta_{ij}} \int_{\Gamma_{ij}^\alpha} \sum_{s=1}^N \mathbf{f}_s(\mathbf{w}) n_s dS \right) dt = 0. \quad (2.12)$$

Zaveďme následující aproximaci:

$$\mathbf{w}_i^k \approx \frac{1}{|D_i|} \int_{D_i} \mathbf{w}(x, t_k) dx, \quad (2.13)$$

kde  $\mathbf{w}_i^k$  nazýváme *přibližným řešením* na objemu  $D_i$  v čase  $t_k$ , a aproximujme tok  $\sum_{s=1}^N \mathbf{f}_s(\mathbf{w})(n_{ij}^\alpha)_s$  veličiny  $\mathbf{w}$  stěnou  $\Gamma_{ij}^\alpha$  pomocí *numerického toku*  $\mathbf{H}(\mathbf{w}_i^k, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha)$ , jež závisí na numerickém řešení  $\mathbf{w}_i^k$  na objemu  $D_i$  a na numerickém řešení  $\mathbf{w}_j^k$  na objemu  $D_j$  a vnější normále  $\mathbf{n}_{ij}^\alpha$  ke stěně  $\Gamma_{ij}^\alpha$  v čase  $t_k$ :

$$\sum_{s=1}^N \mathbf{f}_s(\mathbf{w})(n_{ij}^\alpha)_s \approx \mathbf{H}(\mathbf{w}_i^k, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha). \quad (2.14)$$

Dosazením (2.13) a (2.14) do (2.12) a provedením naznačených aproximací získáme obecné schéma *metody konečných objemů*:

$$\begin{aligned} \mathbf{w}_i^{k+1} &= \mathbf{w}_i^k - \frac{\tau_k}{|D_i|} \sum_{j \in S(i)} \sum_{\alpha=1}^{\beta_{ij}} [\vartheta \mathbf{H}(\mathbf{w}_i^{k+1}, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha) + \\ & (1 - \vartheta) \mathbf{H}(\mathbf{w}_i^k, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha)] |\Gamma_{ij}^\alpha|, \quad D_i \in \mathcal{D}_h, t_k \in [0, T], \vartheta \in [0, 1]. \end{aligned} \quad (2.15)$$

Toto schéma je pro  $\vartheta \in (0, 1]$  *implicitní* a vede k řešení systému obecně nelineárních rovnic pro určení hodnot  $\mathbf{w}_i^{k+1}$ ,  $i \in J$ . Pro hodnotu  $\vartheta = 0$  získáme *explicitní schéma*<sup>3</sup>:

$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - \frac{\tau_k}{|D_i|} \sum_{j \in S(i)} \sum_{\alpha=1}^{\beta_{ij}} \mathbf{H}(\mathbf{w}_i^k, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha) |\Gamma_{ij}^\alpha|, \quad D_i \in \mathcal{D}_h, \quad t_k \in [0, T), \quad (2.16)$$

tedy hodnota  $\mathbf{w}_i^{k+1}$ ,  $i \in J$  se vypočte přímo z hodnot na předcházející časové vrstvě.

Počáteční podmínky  $\mathbf{w}_i^0$ ,  $i \in J$  jsou za předpokladu  $\mathbf{w}^0 \in L_{loc}^1(\Omega)^m$  dány vztahem:

$$\mathbf{w}_i^0 = \frac{1}{|D_i|} \int_{D_i} \mathbf{w}^0(x) dx. \quad (2.17)$$

**Definice 2.1.** *Přibližným řešením metody konečných objemů problému (1.4) budeme rozumět po částech konstantní vektorovou funkci  $\mathbf{w}_h^k$ ,  $k = 0, 1, \dots$ , definovanou skoro všude v  $\Omega_h$  tak, že  $\mathbf{w}_h^k|_{\mathring{D}_i} = \mathbf{w}_i^k$  pro všechna  $i \in J$ , kde  $\mathring{D}_i$  je vnitřek objemu  $D_i$ , tj.  $\mathring{D}_i = D_i \setminus \partial D_i$  a  $\mathbf{w}_i^k$  je dáno vztahem (2.16). Funkce  $\mathbf{w}_h^k$  je aproximace řešení v čase  $t_k$  a vektor  $\mathbf{w}_i^k$  je hodnota aproximace řešení na konečném objemu  $D_i$  v čase  $t_k$ .*

## Vlastnosti numerických toků

O numerickém toku  $\mathbf{H}$  budeme předpokládat, že má následující vlastnosti:

1.  $\mathbf{H}(\mathbf{u}, \mathbf{v}, \mathbf{n})$  je definovaný a spojitý na  $D \times D \times S_1$ , kde  $D$  je definiční obor toků  $\mathbf{f}_s$  a  $S_1$  je jednotková koule v  $\mathbb{R}^N$ :  $S_1 = \{\mathbf{n} \in \mathbb{R}^N; |\mathbf{n}| = 1\}$ .
2.  $\mathbf{H}$  je konzistentní:  

$$\mathbf{H}(\mathbf{u}, \mathbf{u}, \mathbf{n}) = \mathcal{P}(\mathbf{u}, \mathbf{n}) = \sum_{s=1}^N \mathbf{f}_s(\mathbf{u}) n_s, \quad \mathbf{u} \in D, \quad \mathbf{n} \in S_1.$$
3.  $\mathbf{H}$  je konzervativní:  

$$\mathbf{H}(\mathbf{u}, \mathbf{v}, \mathbf{n}) = -\mathbf{H}(\mathbf{v}, \mathbf{u}, \mathbf{n}), \quad \mathbf{u}, \mathbf{v} \in D, \quad \mathbf{n} \in S_1.$$

O metodě konečných objemů říkáme, že je konzistentní, splňuje-li numerický tok  $\mathbf{H}$  metody podmínku konzistence, a že je konzervativní, splňuje-li numerický tok podmínku konzervativnosti.

### 2.2.1 Schéma typu ADER

Schéματα typu *ADER* (Arbitrary Derivative) (viz [45, 46, 42, 23]) jsou schémata, kde se při konstrukci numerického toku (2.14)  $\mathbf{H}(\mathbf{w}_i^k, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha)$  použijí nikoliv vektory po částech konstantních řešení  $\mathbf{w}_i^k$  a  $\mathbf{w}_j^k$ , nýbrž vektory polynomů  $\hat{\mathbf{w}}_i^k$  a  $\hat{\mathbf{w}}_j^k$  v proměných  $x_1, \dots, x_N$ . Tyto polynomy stupně nejvýše  $p - 1$  se získají

---

<sup>3</sup>V celé naší práci byla použita pouze explicitní schémata, proto bude-li v následujícím použito vazby schéma metody konečných objemů, bude myšleno explicitní schéma metody konečných objemů.

vhodnou konstrukcí z po částech konstantních řešení  $\mathbf{w}_i^k$ ,  $i \in J$  a požadujeme od nich následující:

$$\begin{aligned} \frac{1}{|D_i|} \int_{D_i} \hat{\mathbf{w}}_i^k dx &= \mathbf{w}_i^k, \\ \hat{\mathbf{w}}_i^k &= \mathbf{w}(\mathbf{x}, t)|_{D_i} + O(h^p), \quad \mathbf{w} \in C^p(\Omega), \end{aligned} \quad (2.18)$$

kde  $h = \max_{i \in J} \text{diam}(D_i)$ . Popis námi použité rekonstrukce pro  $p = 2$  je uveden v kapitole 2.6.5.

Vyjdeme-li při odvození metody konečných objemů ze vztahu (2.12), můžeme definovat *zobecněný numerický tok*  $\hat{H}$  aproximující integrální průměry toku  $\sum_{s=1}^N \mathbf{f}_s(\mathbf{w})(n_{ij}^\alpha)_s$  přes množinu  $\Gamma_{ij}^\alpha \times (t_k, t_{k+1})$  následovně:

$$\frac{1}{\tau_k} \frac{1}{|\Gamma_{ij}^\alpha|} \int_{t_k}^{t_{k+1}} \int_{\Gamma_{ij}^\alpha} \sum_{s=1}^N \mathbf{f}_s(\mathbf{w})(n_{ij}^\alpha)_s dS dt \approx \hat{H}(\hat{\mathbf{w}}_i^k, \hat{\mathbf{w}}_j^k, \mathbf{n}_{ij}^\alpha). \quad (2.19)$$

Při použití takto definovaného numerického toku přejde explicitní schéma metody konečných objemů (2.16) na tvar:

$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - \frac{\tau_k}{|D_i|} \sum_{j \in S(i)} \sum_{\alpha=1}^{\beta_{ij}} \hat{H}(\hat{\mathbf{w}}_i^k, \hat{\mathbf{w}}_j^k, \mathbf{n}_{ij}^\alpha) |\Gamma_{ij}^\alpha|, \quad D_i \in \mathcal{D}_h, \quad t_k \in [0, T). \quad (2.20)$$

## 2.2.2 Metoda konečných objemů v 1D

Uvažujme Eulerovy rovnice (1.17) pro  $N = 1$ , tedy uvažovaná oblast  $\Omega$  je interval  $[a, b]$ . Pomocí  $x_i$ ,  $i \in Z^+$  označme síťové body, pro tyto body platí  $a = x_0 < x_1 \dots < x_n = b$ . Každý konečný objem  $D_i$  je tvořen síťovými body  $x_i$  a  $x_{i+1}$ , tedy  $D_i = [x_i, x_{i+1}]$ . Konečné objemy tvoří síť konečných objemů  $\mathcal{D}_h = \{D_i\}_{i \in J}$ , kde  $J$  je počet konečných objemů dané sítě konečných objemů.

Středy konečných objemů  $D_i$  označme jako  $x_{i+\frac{1}{2}}$ . Dále pomocí  $h_i$  označme velikost konečného objemu  $D_i$ ,  $h_i = x_{i+1} - x_i$ . Budeme-li uvažovat opět časové dělení  $0 = t_0 < t_1 < \dots$  a časový krok  $\tau_k = t_{k+1} - t_k$ , pak časoprostorovou integrací přes body  $t_k, t_{k+1}$  a body  $x_i, x_{i+1}$  a užitím aproximace:

$$\int_{x_i}^{x_{i+1}} \mathbf{w}(x, t_k) dx \approx h_i \mathbf{w}_i^k, \quad (2.21)$$

a aproximací toků  $\mathbf{f}_1(\mathbf{w}(x_i, t_k))$  v bodech  $x_i$  (viz 2.14), pomocí 1D numerických toků  $\mathbf{g}_i^k$ ,  $\mathbf{f}_1(\mathbf{w}(x_i, t_k)) \approx \mathbf{g}_i^k(\mathbf{w}_{i-1}, \mathbf{w}_i)$ , obdržíme jednorozměrnou analogii explicitního schématu (2.16)[13]:

$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - \frac{\tau_k}{h_i} (\mathbf{g}_{i+1}^k - \mathbf{g}_i^k), \quad i \in Z, \quad k \in Z^+. \quad (2.22)$$

Počáteční řešení  $\mathbf{w}_i^0$  je dáno užitím aproximace (2.21):

$$\mathbf{w}_i^0 = \frac{1}{h_i} \int_{x_i}^{x_{i+1}} \mathbf{w}^0(x) dx. \quad (2.23)$$

## 2.3 Podmínka stability

Pomocí  $\|\mathbf{w}^k\|$  označme nějakou normu přibližného řešení  $\mathbf{w}^k$ , viz definice 2.1. O schématu (2.16) říkáme že je *stabilní*, existuje-li taková konstanta  $c > 0$  nezávislá na  $\tau_k, h, k$ , že platí [13]:

$$\|\mathbf{w}^k\| \leq c\|\mathbf{w}^0\|, \quad k = 0, 1, \dots \quad (2.24)$$

Obvykle se jako normy používají analogie k  $L^p$ -normě ( $p \in [1, \infty]$ ):

$$\|\mathbf{w}^k\|_\infty = \sup_{i \in J} |\mathbf{w}_i^k| \quad (2.25)$$

a

$$\|\mathbf{w}^k\|_p = \left\{ \sum_{i \in J} |D_i| |\mathbf{w}_i^k|^p \right\}^{1/p}, \quad p \in [1, \infty). \quad (2.26)$$

Protože používáme explicitní schéma (2.16), je nutné klást jistá omezení na velikost časového kroku  $\tau_k$ . Pro Eulerovy rovnice se používají následující omezení, která jsou heuristicky určena z přesné podmínky stability schématu pro lineární systémy hyperbolických rovnic [13]. Toto omezení se nazývá *CFL podmínkou stability* (Courant-Friedrichs-Lewy):

$$\tau_k \leq \frac{\text{CFL} |D_i|}{\lambda_{i,max}^k |\partial D_i|}, \quad i \in J, \quad (2.27)$$

a

$$\lambda_{i,max}^k = \max_{\substack{r=1,\dots,m, j \in S(i) \\ \alpha=1,\dots,\beta_{ij}}} |\lambda_r^k(\mathbf{w}_{ij}^k, \mathbf{n}_{ij}^\alpha)|, \quad m = N + 2, \quad (2.28)$$

kde hodnoty  $\lambda_r^k(\mathbf{w}_{ij}^k, \mathbf{n}_{ij}^\alpha)$  jsou vlastní čísla matice  $\mathbb{P}(\mathbf{w}_{ij}^k, \mathbf{n}_{ij}^\alpha)$  (viz (1.21)) a hodnota vektoru  $\mathbf{w}_{ij}^k$  obecně závisí na vektorech  $\mathbf{w}_i^k, \mathbf{w}_j^k$  a na konkrétním použitém numerickém toku, např. pro Vijayasundaramovo a Van Leerovo schéma, viz [13] je  $\mathbf{w}_{ij}^k = \frac{\mathbf{w}_i^k + \mathbf{w}_j^k}{2}$ , CFL je číslo, pro které platí:  $\text{CFL} \in (0, 1)$ .

## 2.4 Riemannův problém

V této části naší práce si ve stručnosti popíšeme *Riemannův problém* a *zobecněný Riemannův problém*, které se používají pro konstrukci některých numerických toků (*metody Godunova typu*, viz kapitola 2.5). Riemannův problém je následující úloha. Hledáme  $\mathbf{w} : \mathbb{R} \times \langle 0, \infty \rangle \mapsto \mathbb{R}^m$  splňující:

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}_1(\mathbf{w})}{\partial x_1} = 0, \quad x_1 \in \mathbb{R}, t > 0, \quad (2.29)$$

za počátečních podmínek:

$$\mathbf{w}(x_1, 0) = \begin{cases} \mathbf{w}_L, & x_1 < 0, \\ \mathbf{w}_R, & x_1 > 0, \end{cases} \quad (2.30)$$

kde  $\mathbf{w}_L, \mathbf{w}_R \in D$  jsou konstanty,  $D$  je dáno (1.16). O tvaru řešení Riemannova problému nám hovoří následující věta [13, Theorem 2.24](důkaz tamtéž):

**Věta 2.1.** *Má-li Riemannův problém jediné po částech hladké řešení  $\mathbf{w}$ , pak  $\mathbf{w}$  může být pro  $t > 0$  zapsáno v podobnostním tvaru:*

$$\mathbf{w}(x_1, t) = \tilde{\mathbf{w}}(x_1/t),$$

kde  $\tilde{\mathbf{w}} : \mathbb{R} \rightarrow \mathbb{R}^m$ .

Navíc takové řešení závisí i na vektorech  $\mathbf{w}_L$ ,  $\mathbf{w}_R$  a můžeme ho psát ve tvaru:

$$\mathbf{w}(x_1, t) = \mathbf{w}_{RS}(x_1/t; \mathbf{w}_L, \mathbf{w}_R), \quad (2.31)$$

kde zkratka RS znamená *Riemann solution*.

V dalším označme  $\lambda_k(\mathbf{w})$ , viz (1.37), vlastní číslo matice  $\mathbb{A}_1(\mathbf{w})$  (viz (1.19)) odpovídající vlastnímu vektoru  $\mathbf{r}_k(\mathbf{w})$  této matice. Zavedme následující pojmy [13, Definition 2.25]:

**Definice 2.2.** *Vlastní vektor  $\mathbf{r}_k = \mathbf{r}_k(\mathbf{w})$  matice  $\mathbb{A}_1 = \mathbb{A}_1(\mathbf{w})$  se nazývá ‘genuinely’ nelineární, jestliže:*

$$\nabla_{\mathbf{w}} \lambda_k(\mathbf{w})^T \cdot \mathbf{r}_k(\mathbf{w}) \neq 0 \quad \forall \mathbf{w} \in D. \quad (2.32)$$

Dále říkáme, že vektor  $\mathbf{r}_k$  je lineárně degenerovaný, když:

$$\nabla_{\mathbf{w}} \lambda_k(\mathbf{w})^T \cdot \mathbf{r}_k(\mathbf{w}) = 0 \quad \forall \mathbf{w} \in D. \quad (2.33)$$

Úpravou věty [13, Theorem 2.32] vztahující se k obecnému hyperbolickému problému, pro náš případ Eulerových rovnic, dostáváme:

**Věta 2.2.** *Předpokládejme, že pro každé  $\mathbf{w} \in D$  všechna vlastní čísla  $\lambda_k(\mathbf{w})$  matice  $\mathbb{A}_1(\mathbf{w})$  jsou jednoduchá a že každý odpovídající vlastní vektor  $\mathbf{r}_k$  je buď ‘genuinely’ nelineární nebo lineárně degenerovaný. Pak pro libovolné  $\mathbf{w}_L \in D$  existuje jeho okolí  $B(\mathbf{w}_L) \in D$  takové, že platí: pro každé  $\mathbf{w}_R \in B(\mathbf{w}_L)$  má Riemannův problém jediné řešení. Toto řešení se skládá z maximálně  $m + 1$  konstantních částí ( $m=N+2$ ).*

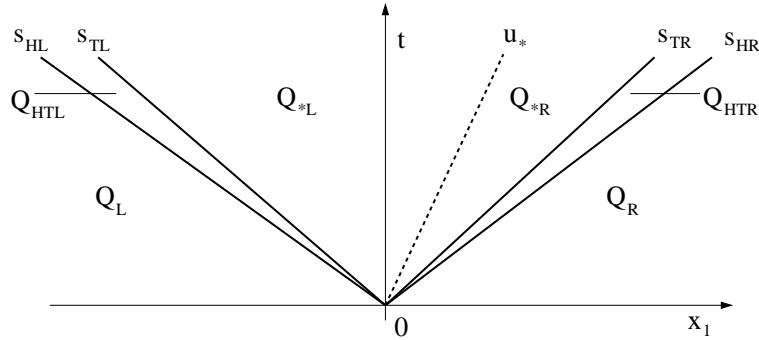
Toto je ilustrováno na následujícím obrázku 2.5, kde je oblast, na níž hledáme řešení Riemannova problému, rozdělena na jednotlivé klíny Q:

$$\begin{aligned} Q_L &= \left\{ (x_1, t); \frac{x_1}{t} < s_{HL}, t > 0 \right\}, \\ Q_{HTL} &= \left\{ (x_1, t); s_{HL} < \frac{x_1}{t} < s_{TL}, t > 0 \right\}, \\ Q_{*L} &= \left\{ (x_1, t); s_{TL} < \frac{x_1}{t} < u_*, t > 0 \right\}, \\ Q_{*R} &= \left\{ (x_1, t); u_* < \frac{x_1}{t} < s_{TR}, t > 0 \right\}, \\ Q_{HTR} &= \left\{ (x_1, t); s_{TR} < \frac{x_1}{t} < s_{HR}, t > 0 \right\}, \\ Q_R &= \left\{ (x_1, t); s_{HR} < \frac{x_1}{t}, t > 0 \right\}. \end{aligned}$$

Množiny  $\overline{Q}_{HTL}$ , resp.  $\overline{Q}_{HTR}$  se nazývají *vlny zředění* a mohou v závislosti na počáteční podmínce degenerovat na množiny  $\{(x_1, t); \frac{x_1}{t} = s_1, t > 0\}$ , respektive  $\{(x_1, t); \frac{x_1}{t} = s_3, t > 0\}$ , v tomto případě je pak nazýváme *rázovými vlnami*.

Množina  $\{(x_1, t); \frac{x_1}{t} = u_*, t > 0\}$  se nazývá *kontaktní nespojitostí*. Konstanty  $s_{HL}, s_{TL}, s_1, u_*, s_3, s_{TR}, s_{HR}$  jsou rychlosti vln. Struktura řešení Riemannova problému na jednotlivých klínech je potom následující:

$$\begin{aligned} \mathbf{w}|_{Q_L} &= \mathbf{w}_L, & \mathbf{w}|_{Q_{*R}} &= \mathbf{w}_{*R}, \\ \mathbf{w}|_{Q_{HTL}} &= \text{hladká funkce}, & \mathbf{w}|_{Q_{HTR}} &= \text{hladká funkce}, \\ \mathbf{w}|_{Q_{*L}} &= \mathbf{w}_{*L}, & \mathbf{w}|_{Q_R} &= \mathbf{w}_R. \end{aligned}$$



Obrázek 2.5: Struktura řešení Riemannova problému. Pomocí tučných čar jsou naznačeny hranice jednotlivých klínů  $Q$ .

## 2.4.1 Přibližný Riemannův řešič HLLC

Námi použitá implementace metody ADER používá, místo přesného řešení Riemannova problému, pouze přibližného Riemannova řešiče HLLC<sup>4</sup>, jehož odvození není součástí této práce. Poznamenejme jen, že místo struktury řešení Riemannova problému zanesené na obrázku 2.5 je použita aproximace řešení problému, skládající se ze čtyř oblastí, na kterých jsou konstantní řešení  $\mathbf{w}_L, \mathbf{w}_{*L}, \mathbf{w}_{*R}$  a  $\mathbf{w}_R$ , viz obrázek 2.6.

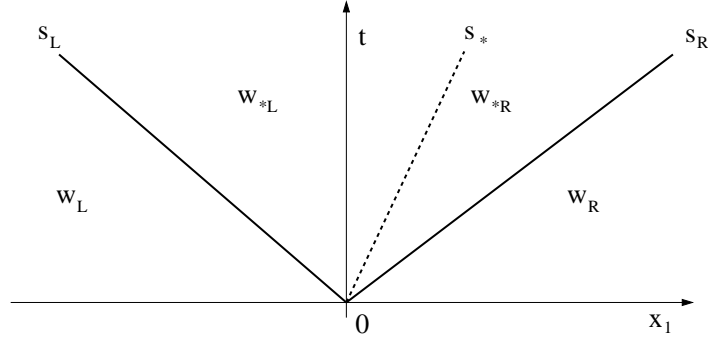
Tato řešení jsou popsána vztahy [23]:

$$\mathbf{w}(x, t) = \begin{cases} \mathbf{w}_L & \text{pro } \frac{x}{t} < s_L, \\ \mathbf{w}_{*L} & \text{pro } s_L < \frac{x}{t} < s_*, \\ \mathbf{w}_{*R} & \text{pro } s_* < \frac{x}{t} < s_R, \\ \mathbf{w}_R & \text{pro } \frac{x}{t} > s_R, \end{cases} \quad (2.34)$$

kde  $s_L, s_*, s_R$  jsou rychlosti vln oddělující jednotlivá po částech konstantní řešení  $\mathbf{w}_L, \mathbf{w}_{*L}, \mathbf{w}_{*R}$  a  $\mathbf{w}_R$ . Řešení  $\mathbf{w}_{*L}, \mathbf{w}_{*R}$  tvořící aproximaci středového klínu

<sup>4</sup>Jde o modifikaci tzv. HLL schématu navrženého autory Hartenem, Laxem, Van Leerem, a navíc obsahující vlnu kontaktní nespojitosti (Contact) ve středovém klínu řešení.





Obrázek 2.6: Struktura řešení přibližného řešení Riemannova problému pro HLLC řešič.

přesného řešení Riemannova problému jsou dána:

$$\mathbf{w}_{*K} = \rho_K \left( \frac{s_K - u_K}{s_K - s_*} \right) \begin{pmatrix} 1 \\ s_* \\ v_K \\ w_K \\ \frac{E_K}{\rho_K} + (s_* - u_K) \left[ s_* + \frac{p_K}{\rho_K(s_K - u_K)} \right] \end{pmatrix}, \quad (2.35)$$

kde  $K = L, R$ . Odhady rychlosti vln  $s_L$ , resp.  $s_R$  jsou určeny jako nejmenší, resp. největší vlastní čísla Roeovy matice  $\hat{\mathbf{A}}(\mathbf{w}_L, \mathbf{w}_R)$  (viz 2.97), kde uvažujeme 1D případ, tedy:

$$s_L = \hat{u} - \hat{a}, \quad s_R = \hat{u} + \hat{a} \quad (2.36)$$

kde  $\hat{u}$  a  $\hat{a}$  jsou dány vztahy (2.93) a (2.98). Hodnotu rychlosti  $s_*$ , která odpovídá rychlosti kontaktní nespojitosti, určíme pomocí [23]:

$$s_* = \frac{p_R - p_L + \rho_L u_L (s_L - u_L) - \rho_R u_R (s_R - u_R)}{\rho_L (s_L - u_L) - \rho_R (s_R - u_R)}. \quad (2.37)$$

## 2.4.2 Zobecněný Riemannův problém a jeho řešení

Podobně jako výše uvedený Riemannův problém, je řešení *zobecněného Riemannova problému (DRP)*<sup>5</sup> využito při konstrukci schématu typu ADER, viz kapitola 2.2.1, které použijeme v numerických experimentech v kapitole 4.

Zobecněným Riemannovým problémem rozumíme problém typu (2.29), kde vektory v počátečních podmínkách (2.30) nejsou konstanty, nýbrž dostatečně hladké funkce a navíc  $\mathbf{w} = \mathbf{w}(\mathbf{x}, t)$ ,  $\mathbf{x} \in \mathbb{R}^N$ , kde  $N = 2$ , nebo  $N = 3$ :

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}_1(\mathbf{w})}{\partial x_1} = 0, \quad x_1 \in \mathbb{R}, t > 0, \quad (2.38)$$

$$\mathbf{w}(\mathbf{x}, 0) = \begin{cases} \hat{\mathbf{w}}_L(\mathbf{x}), & x_1 < 0, \\ \hat{\mathbf{w}}_R(\mathbf{x}), & x_1 > 0. \end{cases} \quad (2.39)$$

<sup>5</sup>Někteří autoři, např. Toro a Titarev v [45, 46, 44, 42], rozlišují mezi označením *GRP* (Generalized Riemann Problem) a *DRP* (Derivative Riemann Problem). V GRP se uvažují po částech lineární počáteční podmínky, zatímco v DRP se uvažují obecné, dostatečně hladké funkce.

V dalším, z důvodu konstrukce numerických toků vyšších řádů, viz kapitola 2.5.1, uvažujeme, že  $\hat{\mathbf{w}}_L(\mathbf{x})$ ,  $\hat{\mathbf{w}}_R(\mathbf{x})$  jsou polynomy stupně nejvýše  $p - 1$ . Tento problém budeme značit jako  $\text{DRP}_{p-1}$ .

Pro naše účely, tedy konstrukci numerických toků na hranici mezi dvěma objemy  $D_i$  a  $D_j$ , se omezme na řešení v bodě  $\mathbf{x}_0 : (\mathbf{x}_0)_1 = 0$ . Provedeme-li Taylorův rozvoj v čase, lze toto řešení aproximovat:

$$\mathbf{w}(\mathbf{x}_0, \tau) \approx \mathbf{w}^+(\mathbf{x}_0, 0) + \sum_{l=1}^{p-1} \left[ \frac{\partial^l}{\partial t^l} \mathbf{w}^+(\mathbf{x}_0, 0) \right] \frac{\tau^l}{l!}, \quad (2.40)$$

kde  $\mathbf{w}^+(\mathbf{x}_0, 0) = \lim_{t \rightarrow 0^+} \mathbf{w}(\mathbf{x}_0, t)$ .

První člen  $\mathbf{w}^+(\mathbf{x}_0, 0)$  v (2.40) se určí řešením Riemannova problému s počáteční podmínkou:

$$\mathbf{w}(x_1, 0) = \begin{cases} \hat{\mathbf{w}}_L(\mathbf{x}_0), & x_1 < 0, \\ \hat{\mathbf{w}}_R(\mathbf{x}_0), & x_1 > 0. \end{cases} \quad (2.41)$$

Toto řešení můžeme zapsat v následujícím podobnostním tvaru:

$$\mathbf{w}^+(\mathbf{x}_0, 0) = \mathbf{w}_{RS}(0; \hat{\mathbf{w}}_L(\mathbf{x}_0), \hat{\mathbf{w}}_R(\mathbf{x}_0)). \quad (2.42)$$

Členy vyšších řádů v rozvoji (2.40) jsou určeny ve dvou krocích. Nejprve pomocí tzv. *procedury Cauchy-Kowalewské* vyjádříme časové derivace  $\frac{\partial^{(l)} \mathbf{w}}{\partial t^l}$   $l = 1, \dots, p - 1$  z rovnice (1.17) pomocí derivací prostorových. V dalším kroku si tyto prostorové derivace spočteme, a to tak, že derivováním vztahu (2.38) podle proměnných  $x_1, \dots, x_N$  až do řádu  $p$  a po úpravách dostáváme evoluční rovnice tohoto tvaru:

$$\frac{\partial \mathbf{w}^{(l_1, \dots, l_N)}}{\partial t} + \mathbb{A}_1(\mathbf{w}) \frac{\partial \mathbf{w}^{(l_1, \dots, l_N)}}{\partial x_1} = \mathbf{H}(\mathbf{w}, \mathbf{w}^{(1,0, \dots, 0)}, \dots, \mathbf{w}^{(l_1, \dots, l_N)}), \quad (2.43)$$

kde

$$\mathbf{w}^{(l_1, \dots, l_N)} = \frac{\partial^{l_1 + \dots + l_N} \mathbf{w}}{\partial^{l_1} x_1 \dots \partial^{l_N} x_N}, \quad 1 \leq l_1 + \dots + l_N \leq p - 1,$$

a  $\mathbf{H}$  je nelineární zdrojový člen. Protože nás zajímají prostorové derivace v bodě  $\mathbf{x}_0$  a v čase  $t \rightarrow 0^+$ , člen  $\mathbf{H}$  se dle [45, 46, 44, 42] zanedbá a dále se rovnice (2.43) linearizuje okolo vedoucího členu  $\mathbf{w}^+(\mathbf{x}_0, 0)$  rozvoje (2.40). Tedy místo matice  $\mathbb{A}_1(\mathbf{w})$  budeme uvažovat konstantní matici:

$$\mathbb{A}_0 = \mathbb{A}_1(\mathbf{w}^+(\mathbf{x}_0, 0)). \quad (2.44)$$

Pro určení prostorových derivací  $\mathbf{w}^{(k_1, \dots, k_N)}$  řešíme sekvenci následujících linearizovaných Riemannových problémů, kde počáteční podmínky jsou určeny extrapolací vektorů  $\hat{\mathbf{w}}_L^{(k_1, \dots, k_N)}$  a  $\hat{\mathbf{w}}_R^{(k_1, \dots, k_N)}$  do bodu  $\mathbf{x}_0$ :

$$\frac{\partial \mathbf{w}^{(k_1, \dots, k_N)}}{\partial t} + \mathbb{A}_0 \frac{\partial \mathbf{w}^{(k_1, \dots, k_N)}}{\partial x_1} = 0 \quad (2.45)$$

$$\mathbf{w}^{(k_1, \dots, k_N)}(x_1, 0) = \begin{cases} \hat{\mathbf{w}}_L^{(k_1, \dots, k_N)}(\mathbf{x}_0), & x_1 < 0, \\ \hat{\mathbf{w}}_R^{(k_1, \dots, k_N)}(\mathbf{x}_0), & x_1 > 0. \end{cases} \quad (2.46)$$

Po řešení problému (2.45) použijeme spočtené prostorové derivace  $\mathbf{w}^{(k_1, \dots, k_N)}$  pro konstrukci časových derivací pomocí Cauchy-Kowalewské procedury a tyto

dosadíme do rozvoje (2.40) a řešení zobecněného Riemannova problému pomocí něho aproximujeme:

$$\hat{\mathbf{w}}_{RS}(\mathbf{x}_0, t; \hat{\mathbf{w}}_L, \hat{\mathbf{w}}_R) \approx \mathbf{w}(\mathbf{x}_0, t). \quad (2.47)$$

Opět jako v případě Riemannova problému poznamenejme, že místo přesného řešení Riemannových problémů pro konstrukci, jak vedoucího členu  $\mathbf{w}^+(\mathbf{x}_0, 0)$ , tak i prostorových derivací  $\mathbf{w}^{(k_1, \dots, k_N)}$ , je možné použít některé z přibližných řešičů Riemannova problému, viz kapitola 2.4.1.

## 2.5 Metody Godunova typu

Metody *Godunova typu*, viz [13], jsou založeny na řešení Riemannova problému (viz kapitola 2.4). Godunova metoda v 1D je definována jako numerické schéma s následujícím numerickým tokem  $\mathbf{g}$ :

$$\mathbf{g}_G(\mathbf{u}, \mathbf{v}) = \mathbf{f}_1(\mathbf{w}_{RS}(0; \mathbf{u}, \mathbf{v})), \quad (2.48)$$

kde funkce  $\mathbf{g}_G$  se nazývá *Godunovův numerický tok*, nebo *přesný Riemannův řešič*<sup>6</sup>. A tedy výsledné 1D schéma je:

$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - \frac{\tau_k}{h_i} [\mathbf{f}_1(\mathbf{w}_{RS}(0; \mathbf{w}_i^k, \mathbf{w}_{i+1}^k)) - \mathbf{f}_1(\mathbf{w}_{RS}(0; \mathbf{w}_{i-1}^k, \mathbf{w}_i^k))]. \quad (2.49)$$

Pro více dimenzí postupujeme následovně. Nechť  $\Gamma_{ij}^\alpha$  je stěna mezi objemy  $D_i$  a  $D_j$  s normálou  $\mathbf{n}_{ij}^\alpha$  směřující z  $D_i$  do  $D_j$ . Zavedme v  $\mathbb{R}^N$ ,  $N = 2$  nebo  $3$  nový kartézský souřadný systém  $\tilde{x}_1, \dots, \tilde{x}_N$  s počátkem ve středu  $\Gamma_{ij}^\alpha$  a s osou  $\tilde{x}_1$  orientovanou ve směru normály  $\mathbf{n}_{ij}^\alpha$  a s  $\tilde{x}_2, \dots, \tilde{x}_N$  tečnými ke stěně  $\Gamma_{ij}^\alpha$ , který jsme získali posunutím počátku souřadného systému do  $\Gamma_{ij}^\alpha$  a otočením tak, že osa  $\tilde{x}_1$  je orientována ve směru normály  $\mathbf{n}_{ij}^\alpha$ . Pak z rotační invariantnosti Eulerových rovnic (1.32) lze psát:

$$\frac{\partial \mathbf{q}}{\partial t} + \sum_{s=1}^N \frac{\partial \mathbf{f}_s(\mathbf{q})}{\partial \tilde{x}_s} = 0, \quad (2.50)$$

kde

$$\mathbf{q} = \mathbb{Q}(\mathbf{n}_{ij}^\alpha) \mathbf{w}, \quad (2.51)$$

a matice rotace  $\mathbb{Q}(\mathbf{n}_{ij}^\alpha)$  je definována v (1.31). Zanedbáme-li tečné derivace ve směrech  $\tilde{x}_2, \dots, \tilde{x}_N$ , dostáváme následující Riemannův problém:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}_1(\mathbf{q})}{\partial \tilde{x}_1} = 0, \quad \tilde{x}_1 \in \mathbb{R}, t > 0, \quad (2.52)$$

za počátečních podmínek:

$$\mathbf{q}(\tilde{x}_1, 0) = \begin{cases} \mathbb{Q}(\mathbf{n}_{ij}^\alpha) \mathbf{w}_i^k, & \tilde{x}_1 < 0, \\ \mathbb{Q}(\mathbf{n}_{ij}^\alpha) \mathbf{w}_j^k, & \tilde{x}_1 > 0, \end{cases} \quad (2.53)$$

kde  $\mathbf{w}_i^k$  a  $\mathbf{w}_j^k$  jsou řešení na objemech  $D_i$  a  $D_j$ . Použijeme-li Godunovův numerický tok (2.48), lze definovat numerický tok pro vícedimensionální Eulerovy rovnice:

$$\mathbf{H}(\mathbf{w}_i^k, \mathbf{w}_j^k, \mathbf{n}_{ij}^\alpha) = \mathbb{Q}^{-1} \mathbf{g}_G(\mathbb{Q} \mathbf{w}_i^k, \mathbb{Q} \mathbf{w}_j^k), \quad (2.54)$$

---

<sup>6</sup>exact Riemann solver

kde  $\mathbb{Q} = \mathbb{Q}(n_{ij}^\alpha)$ .

Pro použití Godunova schématu je nutné řešit Riemannův problém, což je obecně náročné. Tomuto se lze vyhnout tak, že použijeme pouze jeho přibližné řešení:

$$\mathbf{g}_G(\mathbf{u}, \mathbf{v}) \approx \mathbf{g}_R(\mathbf{u}, \mathbf{v}), \quad (2.55)$$

kde  $\mathbf{g}_R$  značí *Riemannův numerický tok*, neboli *přibližný Riemannův řešič*.

V dalším předpokládejme, že Jakobioho matice  $\mathbb{A}(\mathbf{w}) = \mathbb{A}_1(\mathbf{w})$  (viz (1.19)) pro Riemannův problém ve tvaru:

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbb{A}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x_1} = 0, \quad x \in \mathbb{R}, t > 0, \quad (2.56)$$

je diagonalizovatelná:

$$\mathbb{A}(\mathbf{w}) = \mathbb{T}(\mathbf{w}) \mathbb{\Lambda}(\mathbf{w}) \mathbb{T}^{-1}(\mathbf{w}), \quad (2.57)$$

kde  $\mathbb{T}(\mathbf{w})$  je nesingulární matice a  $\mathbb{\Lambda}(\mathbf{w}) = \text{diag}(\lambda_1(\mathbf{w}), \dots, \lambda_m(\mathbf{w}))$  je diagonální maticí vlastních čísel  $\lambda$ . Zavedme označení  $a^+$  resp.  $a^-$  pro libovolné  $a \in \mathbb{R}$ , kde platí  $a^+ = \max(a, 0)$ , resp.  $a^- = \min(a, 0)$  a definujme matice

$$\mathbb{\Lambda}^\pm(\mathbf{w}) = \text{diag}(\lambda_1^\pm(\mathbf{w}), \dots, \lambda_m^\pm(\mathbf{w})), \quad |\mathbb{\Lambda}(\mathbf{w})| = \text{diag}(|\lambda_1(\mathbf{w})|, \dots, |\lambda_m(\mathbf{w})|) \quad (2.58)$$

a

$$\mathbb{A}^\pm(\mathbf{w}) = \mathbb{T}(\mathbf{w}) \mathbb{\Lambda}^\pm(\mathbf{w}) \mathbb{T}^{-1}(\mathbf{w}), \quad |\mathbb{A}(\mathbf{w})| = \mathbb{T}(\mathbf{w}) |\mathbb{\Lambda}(\mathbf{w})| \mathbb{T}^{-1}(\mathbf{w}). \quad (2.59)$$

Na základě tvaru přesného Riemannova řešiče pro lineární Riemannův problém typu (2.56), kde matice  $\mathbb{A}$  je konstantní, hledejme heuristicky přibližný Riemannův řešič ve tvaru, [13]:

$$\begin{aligned} \mathbf{g}_R(\mathbf{u}, \mathbf{v}) &= \mathbf{f}(\mathbf{u}) + \int_{\mathbf{u}}^{\mathbf{v}} \mathbb{A}^-(\mathbf{w}) d\mathbf{w} \\ &= \mathbf{f}(\mathbf{v}) - \int_{\mathbf{u}}^{\mathbf{v}} \mathbb{A}^+(\mathbf{w}) d\mathbf{w} \\ &= \frac{1}{2} \left\{ \mathbf{f}(\mathbf{u}) + \mathbf{f}(\mathbf{v}) - \int_{\mathbf{u}}^{\mathbf{v}} |\mathbb{A}(\mathbf{w})| d\mathbf{w} \right\}, \end{aligned} \quad (2.60)$$

kde

$$\int_{\mathbf{u}}^{\mathbf{v}} \mathbb{A}^\pm(\mathbf{w}) d\mathbf{w} = \left( \int_{\mathbf{u}}^{\mathbf{v}} \mathbf{a}_1^\pm(\mathbf{w}) d\mathbf{w}, \dots, \int_{\mathbf{u}}^{\mathbf{v}} \mathbf{a}_m^\pm(\mathbf{w}) d\mathbf{w} \right)^T. \quad (2.61)$$

Ve vztahu (2.61) označuje  $\mathbf{a}_i^\pm(\mathbf{w})$   $i$ -tý řádek matice  $\mathbb{A}^\pm(\mathbf{w})$  a  $\int_{\mathbf{u}}^{\mathbf{v}} \mathbf{a}_i^\pm(\mathbf{w}) d\mathbf{w}$  je křivkový integrál vektorové funkce  $\mathbf{a}_i^\pm : \mathbb{R}^m \rightarrow \mathbb{R}^m$  podél křivky v  $\mathbb{R}^m$  s počátečním bodem  $\mathbf{u}$  a koncovým bodem  $\mathbf{v}$ . Tento integrál je obecně závislý na integrační cestě. Vztah (2.60) lze spočítat buď pomocí vhodné numerické kvadratury, tím dostaneme např. Stegerův-Warmingův, Vijayasundaramův či Van Leerův numerický tok, nebo integrací po vhodné integrační cestě, Osher-Solomonův numerický tok, viz např. [48].

## 2.5.1 Zobecněná Godunova metoda

Zobecněná Godunova metoda vychází při konstrukci numerického toku  $\mathbf{H}$  z řešení zobecněného Riemannova problému, viz kapitola 2.4.2. V jejím případě můžeme postupovat analogicky k předchozímu, tedy zavedeme nový kartézský souřadný systém  $\tilde{x}_1, \dots, \tilde{x}_N$  s počátkem ve středu  $\Gamma_{ij}^\alpha$ , kde  $N = 2$ , nebo  $N = 3$  a s osou  $\tilde{x}_1$  orientovanou ve směru normály  $\mathbf{n}_{ij}^\alpha$  a s  $\tilde{x}_2, \dots, \tilde{x}_N$  tečnými ke stěně  $\mathbf{n}_{ij}^\alpha$  a po transformaci Eulerových rovnic do tohoto souřadného systému opět obdržíme vztahy (2.50) a (2.51).

Za předpokladu, že jsme schopni sestavit polynomiální rekonstrukce  $\hat{\mathbf{w}}_i^k$  a  $\hat{\mathbf{w}}_j^k$ , viz kapitola 2.2.1, pak po zanedbání tečné derivace ve směrech  $\tilde{x}_2, \dots, \tilde{x}_N$  v (2.50) dostáváme následující zobecněný Riemannův problém:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}_1(\mathbf{q})}{\partial \tilde{x}_1} = 0, \quad \tilde{x}_1 \in \mathbb{R}, t > 0, \quad (2.62)$$

$$\mathbf{q}(\tilde{\mathbf{x}}, 0) = \begin{cases} \hat{\mathbf{q}}_L(\tilde{\mathbf{x}}), & \tilde{x}_1 < 0, \\ \hat{\mathbf{q}}_R(\tilde{\mathbf{x}}), & \tilde{x}_1 > 0, \end{cases} \quad (2.63)$$

kde počáteční podmínky  $\hat{\mathbf{q}}_L$  a  $\hat{\mathbf{q}}_R$  jsou polynomy stupně nejvýše  $p - 1$ , které získáme pomocí transformace typu (2.51):

$$\begin{aligned} \hat{\mathbf{q}}_L(\tilde{\mathbf{x}}) &= \mathbb{Q}(\mathbf{n}_{ij}^\alpha) \hat{\mathbf{w}}_i^k(\tilde{\mathbf{x}}) \\ \hat{\mathbf{q}}_R(\tilde{\mathbf{x}}) &= \mathbb{Q}(\mathbf{n}_{ij}^\alpha) \hat{\mathbf{w}}_j^k(\tilde{\mathbf{x}}), \end{aligned} \quad (2.64)$$

kde matice  $\mathbb{Q}(\mathbf{n}_{ij}^\alpha)$  je definována vztahem (1.31). Za předpokladu, že jsme schopni zkonstruovat řešení  $\hat{\mathbf{q}}_{RS}$ , viz kapitola 2.4.2, tohoto zobecněného Riemannova problému, lze definovat *zobecněný Godunovův tok*:

$$\hat{\mathbf{g}}_G(\hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R) = \frac{1}{\tau_k} \frac{1}{|\Gamma_{ij}^\alpha|} \int_{t_k}^{t_{k+1}} \int_{\tilde{\Gamma}_{ij}^\alpha} \mathbf{f}_1(\hat{\mathbf{q}}_{RS}(\tilde{\mathbf{x}}, t; \hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R)) dS dt, \quad (2.65)$$

kde  $\tilde{\Gamma}_{ij}^\alpha$  je dána transformací hrany, resp. stěny  $\Gamma_{ij}^\alpha$  pro  $N = 2$ , resp.  $N = 3$  do výše uvedeného souřadného systému. Zobecněný numerický tok (2.19) lze psát:

$$\hat{\mathbf{H}}(\hat{\mathbf{w}}_i^k, \hat{\mathbf{w}}_j^k, \mathbf{n}_{ij}^\alpha) = \mathbb{Q}^{-1} \hat{\mathbf{g}}_G(\hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R), \quad (2.66)$$

kde  $\mathbb{Q} = \mathbb{Q}(\mathbf{n}_{ij}^\alpha)$ .

Integrál přes hranu  $\tilde{\Gamma}_{ij}^\alpha$  v (2.65) je při praktické realizaci metody aproximován pomocí Gaussovy kvadratury. Obdobně je možné též aproximovat integrál přes časový interval  $[t_k, t_{k+1}]$ . V naší práci používáme implementaci schématu, jež součástí práce [23], která tento integrál aproximuje analyticky pomocí metody Taylorova rozvoje, tento postup je detailněji popsán v kapitole 2.6.5.

## 2.6 Použité numerické toky a schémata

V této kapitole se zaměříme na popis konstrukce těch schémat, která byla použita při našich numerických experimentech v kombinaci s adaptivními metodami. Podstatně obsáhlejší přehled numerických toků, včetně popisů jejich konstrukce, je možné nalézt v [13], odkud jsme čerpali. V dalším budeme ve všech 1D případech používat značení  $\mathbf{f} := \mathbf{f}_1$ , viz vztah (1.14), a  $\mathbb{A} := \mathbb{A}_1$ , viz (1.19).

### 2.6.1 1D Lax-Friedrichsův numerický tok

Lax-Friedrichsovo schéma je jedním z nejjednodušších a nejčastěji používaných schémat. Lax-Friedrichsův numerický tok je definován následovně:

$$\mathbf{g}_{LF}(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \left( \mathbf{f}(\mathbf{u}) + \mathbf{f}(\mathbf{v}) - \frac{h}{\tau}(\mathbf{v} - \mathbf{u}) \right), \quad (2.67)$$

po dosazení  $\mathbf{g} := \mathbf{g}_{LF}$  do obecného tvaru metody konečných objemů v 1D (2.22), budeme-li uvažovat různou velikost  $h_i$  jednotlivých konečných objemů  $D_i$  a proměnný časový krok  $\tau_k$ , obdržíme Lax-Friedrichsovo explicitní schéma:

$$\mathbf{w}_i^{k+1} = \frac{1}{2} (\mathbf{w}_{i+1}^k + \mathbf{w}_{i-1}^k) - \frac{\tau_k}{2h_i} (\mathbf{f}_{i+1}^k - \mathbf{f}_{i-1}^k), \quad i \in Z, k \in Z^+, \quad (2.68)$$

kde  $\mathbf{f}_{i\pm 1}^k = \mathbf{f}(\mathbf{w}_{i\pm 1}^k)$ . Toto schéma je za podmínek, že přesné řešení  $\mathbf{w}$  a tok  $\mathbf{f}$  mají spojitě třetí derivace prvního řádu v prostoru a čase. Schéma je *podmíněně stabilní*, tj. je nutné splnit CFL podmínku, která přejde ze vztahu (2.27) do následujícího tvaru:

$$\tau_k \leq \frac{\text{CFL } h_i}{\sigma(\mathbb{A}(\mathbf{w}_i^k))}, \quad i \in Z, \quad (2.69)$$

kde  $\sigma(\mathbb{A})$  je spektrální poloměr matice  $\mathbb{A}$ , tj. Jakobiánu funkce  $\mathbf{f}(\mathbf{w})$  (1.19).

### 2.6.2 1D MUSCL Lax-Friedrichsův numerický tok

Protože se v naší práci též věnujeme porovnání námi navržených adaptivních algoritmů s tzv. moving mesh algoritmy, uvedenými v práci [40], pro skalární rovnice typu zákonů zachování, tedy speciální případ rovnic (1.17):

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad (2.70)$$

popišme si 1D *MUSCL* (Monotone Upstream-Centered Scheme for Conservation Laws) schéma, které budeme používat pro testy na úlohách typu (2.70).

V práci [40], se kterou porovnáváme námi navržené adaptivní metody, je použita následující definice MUSCL Lax-Friedrichsova numerického toku:

$$g(u, v)_{MLF} = \frac{1}{2} \left( f(u) + f(v) - \max_u \{|f_u|\} (v - u) \right), \quad f_u = \frac{\partial f(u)}{\partial u}, \quad (2.71)$$

kde člen  $\max_u \{|f_u|\}$  má, obdobně jako  $h/\tau$  v toku (2.67), význam rychlosti a slouží zde ke snížení numerické viskozity schématu, viz např. [38], která je díky tomuto členu menší, než pro schéma předchozí, více lze nalézt v práci [31]. Označme si jako  $u_i^k$  skalární aproximaci typu (2.21). Hodnotu  $u_i^{k+1}$  na další časové vrstvě lze spočítat pomocí schématu analogickému k (2.22):

$$u_i^{k+1} = u_i^k - \frac{\tau_k}{h_i} (g_{i+1}^k - g_i^k), \quad i \in Z, k \in Z^+, \quad (2.72)$$

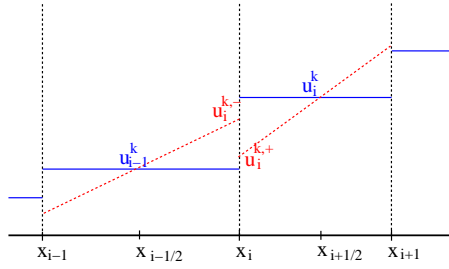
kde

$$\begin{aligned} g_{i+1}^k &= g_{MLF}(u_{i+1}^{k,-}, u_{i+1}^{k,+}), \\ g_i^k &= g_{MLF}(u_i^{k,-}, u_i^{k,+}), \end{aligned} \quad (2.73)$$

jsou výše uvedené Lax-Friedrichsovy numerické toky (2.71), kde jsou z důvodu zvýšení řádu schématu použity lineární rekonstrukce k výpočtu hodnot  $u_i^{k,\pm}$  a  $u_{i+1}^{k,\pm}$  na hranicích příslušného objemu, viz obrázek 2.7:

$$\begin{aligned} u_i^{k,+} &= u_i^k + \frac{1}{2} (x_i - x_{i+1}) \tilde{S}_{i+\frac{1}{2}}, \\ u_i^{k,-} &= u_{i-1}^k + \frac{1}{2} (x_i - x_{i-1}) \tilde{S}_{i-\frac{1}{2}}. \end{aligned} \quad (2.74)$$

Pomocí  $\tilde{S}_{i+\frac{1}{2}}$ , resp.  $\tilde{S}_{i-\frac{1}{2}}$ , je aproximována hodnota směrnice  $\frac{\partial u}{\partial x}$  ve středu  $i$ -tého, resp.  $i-1$  objemu, tento střed označujeme v dalším jako  $x_{i+\frac{1}{2}}$ , resp.  $x_{i-\frac{1}{2}}$ .



Obrázek 2.7: Schéma konstrukce hodnot  $u_i^{k,\pm}$ . Modře jsou naznačeny hodnoty  $u_i^k$ ,  $u_{i-1}^k$ , pomocí červeného čerchování jsou naznačeny směrnice  $\tilde{S}_{i\pm\frac{1}{2}}$  a červeně hodnoty  $u_i^{k,\pm}$ .

Pro výpočet  $\tilde{S}_{i+\frac{1}{2}}$ , resp.  $\tilde{S}_{i-\frac{1}{2}}$ , je v [40] použito následujícího limiteru:

$$\begin{aligned} \tilde{S}_{i+\frac{1}{2}} &= \left( \text{sign}(\tilde{S}_{i+\frac{1}{2}}^+) + \text{sign}(\tilde{S}_{i+\frac{1}{2}}^-) \right) \frac{|\tilde{S}_{i+\frac{1}{2}}^+ \tilde{S}_{i+\frac{1}{2}}^-|}{|\tilde{S}_{i+\frac{1}{2}}^+| + |\tilde{S}_{i+\frac{1}{2}}^-|}, \\ \tilde{S}_{i-\frac{1}{2}} &= \left( \text{sign}(\tilde{S}_{i-\frac{1}{2}}^+) + \text{sign}(\tilde{S}_{i-\frac{1}{2}}^-) \right) \frac{|\tilde{S}_{i-\frac{1}{2}}^+ \tilde{S}_{i-\frac{1}{2}}^-|}{|\tilde{S}_{i-\frac{1}{2}}^+| + |\tilde{S}_{i-\frac{1}{2}}^-|}, \end{aligned} \quad (2.75)$$

kde hodnoty jednotlivých diferencí  $\tilde{S}_{i\pm\frac{1}{2}}^\pm$  jsou dány:

$$\begin{aligned} \tilde{S}_{i+\frac{1}{2}}^+ &= \frac{u_{i+1} - u_i}{x_{i+\frac{3}{2}} - x_{i+\frac{1}{2}}}, & \tilde{S}_{i+\frac{1}{2}}^- &= \frac{u_i - u_{i-1}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}, \\ \tilde{S}_{i-\frac{1}{2}}^+ &= \frac{u_i - u_{i-1}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}, & \tilde{S}_{i-\frac{1}{2}}^- &= \frac{u_{i-1} - u_{i-2}}{x_{i-\frac{1}{2}} - x_{i-\frac{3}{2}}}. \end{aligned} \quad (2.76)$$

Tato metoda je druhého řádu přesnosti v hladkých oblastech. Opět je nutné splnit CFL podmínku (2.69), kde místo Jakobiánu  $\mathbb{A}$ , uvažujeme derivaci  $\frac{\partial f(u)}{\partial u}$ .

### 2.6.3 1D Lax-Wendroffův numerický tok

Za předpokladu, že přesné řešení  $\mathbf{w}$  a tok  $\mathbf{f}$  mají spojitě derivace čtvrtého řádu, lze pomocí Taylorova rozvoje [13] odvodit Lax-Wendroffovo schéma, které je druhého řádu přesnosti v čase a prostoru. Lax-Wendroffův numerický tok je definován:

$$\mathbf{g}_{LW}(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \left( (\mathbf{f}(\mathbf{u}) + \mathbf{f}(\mathbf{v})) - \frac{\tau}{h} \mathbb{A}((\mathbf{u} + \mathbf{v})/2) (\mathbf{f}(\mathbf{v}) - \mathbf{f}(\mathbf{u})) \right). \quad (2.77)$$

Dosazením vztahu (2.77) do (2.22), budeme-li uvažovat různou velikost  $h_i$  jednotlivých konečných objemů  $D_i$  a proměnný časový krok  $\tau_k$ , obdržíme Lax-Wendroffovo explicitní schéma:

$$\begin{aligned} \mathbf{w}_i^{k+1} &= \mathbf{w}_i^k - \frac{\tau_k}{2h_i} (\mathbf{f}_{i+1}^k - \mathbf{f}_{i-1}^k) + \\ &+ \left( \frac{\tau_k}{h_i} \right)^2 \left( \mathbb{A}_{i+\frac{1}{2}}^k (\mathbf{f}_{i+1}^k - \mathbf{f}_i^k) - \mathbb{A}_{i-\frac{1}{2}}^k (\mathbf{f}_i^k - \mathbf{f}_{i-1}^k) \right) / 2, \quad i \in Z, k \in Z^+, \end{aligned} \quad (2.78)$$

kde  $\mathbf{f}_i^k = \mathbf{f}(\mathbf{w}_i^k)$  a  $\mathbb{A}_{i\pm\frac{1}{2}}^k = \mathbb{A}((\mathbf{w}_{i\pm 1}^k + \mathbf{w}_i^k)/2)$ . Opět platí CFL podmínka (2.69).

## 2.6.4 1D Roeův numerický tok

V následujícím textu odvodíme Roeův numerický tok pro  $N = 1$  ( $m = N + 2$ ), budeme postupovat dle [13], kde je tato metoda odvozena pro  $N = 3$ . Roeova metoda spočívá v náhradě Jakobiánu  $\mathbb{A}(\mathbf{w})$  při aproximaci Riemannova problému, viz (2.56), pomocí konstantní *Roeovy matice*  $\tilde{\mathbb{A}} = \tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$ , která závisí pouze na hodnotách stavových vektorů  $\mathbf{w}_L, \mathbf{w}_R$ . Na tuto matici jsou kladeny následující požadavky:

1. Matice  $\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$  je *diagonalizovatelná*:

$$\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R) = \tilde{\mathbb{T}}(\mathbf{w}_L, \mathbf{w}_R) \tilde{\mathbb{\Lambda}}(\mathbf{w}_L, \mathbf{w}_R) \tilde{\mathbb{T}}^{-1}(\mathbf{w}_L, \mathbf{w}_R), \quad (2.79)$$

kde  $\tilde{\mathbb{\Lambda}}(\mathbf{w}_L, \mathbf{w}_R) = \text{diag}(\lambda_1(\mathbf{w}_L, \mathbf{w}_R), \dots, \lambda_m(\mathbf{w}_L, \mathbf{w}_R))$  je diagonální matice vlastních čísel  $\lambda_i(\mathbf{w}_L, \mathbf{w}_R)$ . Sloupce matice  $\tilde{\mathbb{T}}(\mathbf{w}_L, \mathbf{w}_R)$  jsou vlastní vektory  $\mathbf{r}_i(\mathbf{w}_L, \mathbf{w}_R)$  odpovídající vlastním číslům  $\lambda_i(\mathbf{w}_L, \mathbf{w}_R)$ ,  $i = 1, \dots, m$ .

2. Matice  $\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$  je *konzistentní*:

$$\tilde{\mathbb{A}}(\mathbf{w}, \mathbf{w}) = \mathbb{A}(\mathbf{w}). \quad (2.80)$$

3. Matice  $\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$  je *konzervativní* na nespojitostech:

$$\mathbf{f}(\mathbf{w}_R) - \mathbf{f}(\mathbf{w}_L) = \tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R) (\mathbf{w}_R - \mathbf{w}_L). \quad (2.81)$$

Analogicky k (2.58) a (2.59) zavedme značení  $\tilde{\mathbb{A}}^\pm(\mathbf{w}_L, \mathbf{w}_R)$  a aproximujme v přibližném Riemannově řešiči (2.60) matici  $\mathbb{A}^-(\mathbf{w})$  pomocí konstantní matice  $\tilde{\mathbb{A}}^-(\mathbf{w}_L, \mathbf{w}_R)$ , dostáváme:

$$\begin{aligned} \mathbf{g}_{Roe}(\mathbf{w}_L, \mathbf{w}_R) &= \mathbf{f}(\mathbf{w}_L) + \tilde{\mathbb{A}}^-(\mathbf{w}_L, \mathbf{w}_R) \int_{\mathbf{w}_L}^{\mathbf{w}_R} d\mathbf{w} \\ &= \mathbf{f}(\mathbf{w}_L) + \tilde{\mathbb{A}}^-(\mathbf{w}_L, \mathbf{w}_R) (\mathbf{w}_R - \mathbf{w}_L). \end{aligned} \quad (2.82)$$

Při splnění výše uvedeného požadavku diagonalizovatelnosti matice  $\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$ , viz (2.79), je možné vyjádřit stavové vektory  $\mathbf{w}_L$  a  $\mathbf{w}_R$  v bázi tvořené vektory  $\mathbf{r}_i(\mathbf{w}_L, \mathbf{w}_R)$ ,  $i = 1, \dots, m$ :

$$\mathbf{w}_R - \mathbf{w}_L = \sum_{i=1}^m \gamma_i \mathbf{r}_i(\mathbf{w}_L, \mathbf{w}_R). \quad (2.83)$$



Dosazením (2.83) do vztahu (2.82) dostáváme:

$$\begin{aligned} \mathbf{g}_{Roe}(\mathbf{w}_L, \mathbf{w}_R) &= \mathbf{f}(\mathbf{w}_L) + \sum_{i=1}^m \gamma_i \tilde{\mathbb{A}}^-(\mathbf{w}_L, \mathbf{w}_R) \mathbf{r}_i(\mathbf{w}_L, \mathbf{w}_R) \\ &= \mathbf{f}(\mathbf{w}_L) + \sum_{\lambda_i \leq 0} \gamma_i \lambda_i(\mathbf{w}_L, \mathbf{w}_R) \mathbf{r}_i(\mathbf{w}_L, \mathbf{w}_R), \end{aligned} \quad (2.84)$$

což představuje vztah pro vyčíslení Roeova numerického toku v případě, že máme k dispozici vlastní vektory  $\mathbf{r}_i(\mathbf{w}_L, \mathbf{w}_R)$ , vlastní čísla  $\lambda_i(\mathbf{w}_L, \mathbf{w}_R)$ ,  $i = 1, \dots, m$  a koeficienty  $\gamma_i$ .

V dalším textu se zaměříme na konstrukci  $\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$ . Nejprve vyjádříme stavový vektor  $\mathbf{w} = \bar{\mathbf{w}}(\mathbf{z})$  a tok  $\mathbf{f}(\mathbf{w}) = \mathbf{f}(\bar{\mathbf{w}}(\mathbf{z})) = \bar{\mathbf{f}}(\mathbf{z})$  jako kvadratické funkce vektoru  $\mathbf{z}$ :

$$\bar{\mathbf{w}}(\mathbf{z}) = \left( z_1^2, z_1 z_2, \frac{z_1 z_3}{\gamma} + \frac{\gamma - 1}{2\gamma} z_2^2 \right)^T \quad (2.85)$$

a

$$\bar{\mathbf{f}}(\mathbf{z}) = \left( z_1 z_2, z_2^2 + \frac{\gamma - 1}{\gamma} \left[ z_1 z_3 - \frac{z_2^2}{2} \right], z_2 z_3 \right), \quad (2.86)$$

kde význam složek vektoru  $\mathbf{z}$  je následující

$$\mathbf{z} = (z_1, z_2, z_3)^T = \sqrt{\rho} (1, v_1, H)^T, \quad (2.87)$$

kde

$$H = \frac{E + p}{\rho}. \quad (2.88)$$

Na základě lematu uvedeném v [13, Lemma 3.39] a výše uvedeného lze psát:

$$\mathbf{f}(\mathbf{w}_R) - \mathbf{f}(\mathbf{w}_L) = \mathbb{A} \left( \bar{\mathbf{w}} \left( \frac{\mathbf{z}_L + \mathbf{z}_R}{2} \right) \right) (\mathbf{w}_R - \mathbf{w}_L), \quad (2.89)$$

kde hodnoty parametrů  $\mathbf{z}_L$  a  $\mathbf{z}_R$  jsou dány:

$$\mathbf{w}_L = \bar{\mathbf{w}}(\mathbf{z}_L) \quad \text{a} \quad \mathbf{w}_R = \bar{\mathbf{w}}(\mathbf{z}_R). \quad (2.90)$$

Pro splnění požadavku konzervativity matice  $\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R)$  (2.81) ji definujeme následovně:

$$\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R) = \mathbb{A} \left( \bar{\mathbf{w}} \left( \frac{\mathbf{z}_L + \mathbf{z}_R}{2} \right) \right). \quad (2.91)$$

Zavedením Roeových průměrů  $\hat{\rho}, \hat{u}, \hat{H}$  tak, aby platilo:

$$\frac{1}{2} (\mathbf{z}_L + \mathbf{z}_R) = \sqrt{\hat{\rho}} (1, \hat{u}, \hat{H})^T, \quad (2.92)$$

dostáváme pro tyto průměry následující rovnice:

$$\begin{aligned} \sqrt{\hat{\rho}} &= \frac{1}{2} (\sqrt{\rho_L} + \sqrt{\rho_R}), \\ \hat{u} &= \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \hat{H} &= \frac{\sqrt{\rho_L} H_L + \sqrt{\rho_R} H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \end{aligned} \quad (2.93)$$

kde  $\rho_L, \rho_R, u_L, u_R, \dots$  jsou určeny ze  $\mathbf{z}_L$  a  $\mathbf{z}_R$  pomocí (2.87). Použitím vztahů (2.85) a (2.92) vidíme, že platí:

$$\overline{\mathbf{w}} \left( \frac{1}{2} (\mathbf{z}_L + \mathbf{z}_R) \right) = \left( \hat{\rho}, \hat{\rho} \hat{u}, \hat{E} \right)^T = \hat{\mathbf{w}}, \quad (2.94)$$

kde  $\hat{E}$  je dáno

$$\hat{E} = \frac{1}{\gamma} \hat{\rho} \hat{H} + \frac{\gamma - 1}{2\gamma} \hat{\rho} \hat{u}^2. \quad (2.95)$$

Z výše uvedeného je vidět, že konstantní Roeovu matici lze psát ve tvaru:

$$\tilde{\mathbb{A}}(\mathbf{w}_L, \mathbf{w}_R) = \mathbb{A}(\hat{\mathbf{w}}), \quad (2.96)$$

připomeňme si jen, že pomocí  $\mathbb{A}$  značíme matici  $\mathbb{A}_1$ , viz (1.19), pro 1D.

Pro určení vlastních čísel a vlastních vektorů matice  $\mathbb{A}(\hat{\mathbf{w}})$  postupujme následovně. Protože platí vztah (2.96), lze vlastní čísla  $\hat{\lambda}$ , závisující na Roeových průměrech, určit z (1.37), kde uvažujeme 1D případ:

$$\hat{\lambda}_1 = \hat{u} - \hat{a}, \quad \hat{\lambda}_2 = \hat{u}, \quad \hat{\lambda}_3 = \hat{u} + \hat{a}, \quad (2.97)$$

kde

$$\hat{a}^2 = (\gamma - 1) \left( \hat{H} - \frac{\hat{u}^2}{2} \right). \quad (2.98)$$

Obdobně lze vyjít z vlastních vektorů matice  $\mathbb{A}_1(\mathbf{w})$ , např. [13, Lemma 3.3] a po dosazení  $\hat{\mathbf{w}}$  místo  $\mathbf{w}$ , lze pro 1D obdržet:

$$\begin{aligned} \hat{\mathbf{r}}_1 &= \left( 1, \hat{u} - \hat{a}, \hat{H} - \hat{u}\hat{a} \right)^T, \\ \hat{\mathbf{r}}_2 &= \left( 1, \hat{u}, \frac{\hat{u}^2}{2} \right)^T, \\ \hat{\mathbf{r}}_3 &= \left( 1, \hat{u} + \hat{a}, \hat{H} + \hat{u}\hat{a} \right)^T. \end{aligned} \quad (2.99)$$

Koeficienty  $\gamma_i$  obdržíme z rovnice (2.83), kde pro jednoduchost použijeme následující značení  $\Delta w_i = (\mathbf{w}_R)_i - (\mathbf{w}_L)_i$ <sup>7</sup>:

$$\begin{aligned} \gamma_2 &= \frac{\gamma - 1}{\hat{a}^2} \left[ \Delta w_1 \left( \hat{H} - \hat{u}^2 \right) + \hat{u} \Delta w_2 - \Delta w_3 \right], \\ \gamma_1 &= \frac{1}{2\hat{a}} \left[ \Delta w_1 (\hat{u} + \hat{a}) - \Delta w_2 - \hat{a} \gamma_2 \right], \\ \gamma_3 &= \Delta w_1 - (\gamma_1 + \gamma_2). \end{aligned} \quad (2.100)$$

Vyhodnocení Roeova numerického toku probíhá tedy tak, že nejprve dle vzorců (2.93) určíme Roeovy průměry a tyto použijeme pro výpočet vlastních čísel (2.97) a vektorů (2.99). Dále se pak spočtou koeficienty  $\gamma$  dle vzorců (2.100) a na závěr se vyhodnotí Roeův numerický tok (2.84). Schéma je prvního řádu a pro stabilitu metody musí být splněna CFL podmínka (2.69).

<sup>7</sup>Koeficienty zde uvádíme v pořadí, ve kterém je výhodné je vyhodnocovat při praktických výpočtech.

### 2.6.5 Metoda ADER

Zde si popíšeme námi použitou metodu ADER, která je, jak jsme viděli, zobecněním metod Godunova typu. V našich numerických experimentech byla použita implementace této metody pro  $N = 2$  a  $p = 2$ , jež je součástí práce [23].

Budeme používat lineární rekonstrukce  $\hat{\mathbf{w}}_i^k$  pomocí zobecněných centrálních diferencí s limiterem. Rekonstrukce je prováděna po jednotlivých složkách  $u_i$  vektoru řešení  $\mathbf{w}_i^k$  na objemu  $D_i$ . Pro splnění požadavků (2.18) kladených na rekonstrukci  $\hat{\mathbf{w}}_i^k$  budeme hledat rekonstrukci ve tvaru:

$$\hat{u}_i(x_1, x_2) = a(x_1 - x_{1,T_i}) + b(x_2 - x_{2,T_i}) + c, \quad (2.101)$$

kde  $x_{1,T_i}$  a  $x_{2,T_i}$  jsou souřadnice těžiště  $P_i$  objemu  $D_i$ , kde koeficienty  $a$ ,  $b$  představují složky aproximace gradientu  $\nabla u(P_i)$ :

$$\begin{aligned} a &\approx \frac{\partial}{\partial x_1} u(P_i), \\ b &\approx \frac{\partial}{\partial x_2} u(P_i), \\ c &= u_i. \end{aligned} \quad (2.102)$$

Aproximace gradientu, tedy koeficienty  $a$  a  $b$ , určíme jako složky gradientu lineární funkce  $\bar{u}(x_1, x_2)$ , která v těžištích  $P_j$  sousedních objemů  $D_j$  k objemu  $D_i$  nabývá hodnot  $u_j$ . Podmínka

$$\bar{u}(x_{1,T_j}, x_{2,T_j}) = u_j, \quad j \in s(i), \quad (2.103)$$

kde  $x_{1,T_j}$  a  $x_{2,T_j}$  jsou souřadnice těžišť  $P_j$ , představuje soustavu lineárních rovnic, jejímž řešením získáme aproximace složek gradientu. Abychom se vyhnuli nežádoucím oscilacím poblíž nespojitostí, nejprve detekujeme ty objemy, které obsahují nespojitost a položíme na nich gradient nulový ( $a = 0$ ,  $b = 0$ ). Pro nalezení nespojitosti je použit *indikátor skoku* [13, str. 483]:

$$g(i) = \sum_{j \in S(i)} \int_{\Gamma_{ij}} (\hat{u}_i - \hat{u}_j)^2 dS / (h_i |D_i|^{3/4}), \quad (2.104)$$

kde  $\hat{u}_i$  a  $\hat{u}_j$  jsou výše uvedené lineární rekonstrukce a  $h_i = \text{diam}(D_i)$ . Integrál v (2.104) je vyhodnocen pomocí dvoubodového Gaussova kvadraturního pravidla. Výsledný postup pro rekonstrukci s limitingem je tedy následující:

1. Pomocí vztahů ((2.103) a (2.101)) sestrojíme rekonstrukce  $\hat{u}_i$ ,  $i \in J$
2. V případě, že  $g(i) > 1$ , pak  $a, b = 0$  a  $\hat{u}_i = u_i$ .

Jsou-li zkonstruovány polynomiální rekonstrukce  $\hat{\mathbf{w}}_i^k$  a  $\hat{\mathbf{w}}_j^k$ , pak lze spočítat zobecněný Godunovův tok (2.65) a následně vypočítat zobecněný numerický tok (2.66), viz kapitola 2.5.1. Pro vyhodnocení integrálu přes hranici  $\tilde{\Gamma}_{ij}$  v zobecněném Godunovově toku, tedy ve vztahu:

$$\hat{\mathbf{g}}_G(\hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R) = \frac{1}{\tau_k} \frac{1}{|\tilde{\Gamma}_{ij}|} \int_{t_k}^{t_{k+1}} \int_{\tilde{\Gamma}_{ij}} \mathbf{f}_1(\hat{\mathbf{q}}_{RS}(\tilde{\mathbf{x}}, t; \hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R)) dS dt, \quad (2.105)$$

použijeme Gaussovu kvadraturu a obdržíme:

$$\hat{\mathbf{g}}_G(\hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R) = \frac{1}{\tau_k} \int_{t_k}^{t_{k+1}} \left( \sum_{\nu=1}^{K_\nu} \mathbf{f}_1(\hat{\mathbf{q}}_{RS}(\tilde{\mathbf{x}}_\nu, t; \hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R)) w_\nu \right) dt, \quad (2.106)$$

kde  $\tilde{\mathbf{x}}_\nu$  jsou integrační uzly přes hranu  $\tilde{\Gamma}_{ij}$ ,  $K_\nu$  je jejich počet a pomocí  $w_\nu$  jsou označeny příslušné váhy. V implementaci metody, kterou jsme měli k dispozici, bylo  $K_\nu = 1$ . Integrace přes časový interval  $[t_k, t_{k+1}]$  v (2.106) je v práci [23] vyhodnocena analyticky. Nejprve vyjádříme tok  $\mathbf{f}_1$  v kvadrturním bodě  $\mathbf{x}_\nu$  pomocí Taylorova rozvoje:

$$\mathbf{f}_1(\mathbf{q}(\tilde{\mathbf{x}}_\nu, \tau)) = \mathbf{f}_1(\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0)) + \sum_{l=1}^{p-1} \left[ \frac{\partial^l}{\partial t^l} \mathbf{f}_1(\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0)) \right] \frac{\tau^l}{l!}, \quad (2.107)$$

kde  $\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0)$  je vedoucí prvek v rozvoji (2.40) a členy  $\frac{\partial^l}{\partial t^l} \mathbf{f}_1(\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0))$  lze určit pomocí Cauchy-Kowalewské procedury, viz kapitola 2.4.2. Vzhledem k aproximaci (2.47) lze psát:

$$\mathbf{f}_1(\hat{\mathbf{q}}_{RS}(\tilde{\mathbf{x}}_\nu, t, \mathbf{q}_L, \mathbf{q}_R)) \approx \mathbf{f}_1(\mathbf{q}(\tilde{\mathbf{x}}_\nu, t)). \quad (2.108)$$

Dosazením (2.108) a (2.107) do (2.106) a provedením naznačené integrace dle času dostáváme následující vztah pro výpočet zobecněného Godunovova toku:

$$\hat{\mathbf{g}}_G(\hat{\mathbf{q}}_L, \hat{\mathbf{q}}_R) = \sum_{\nu=1}^{K_\nu} \left( \mathbf{f}_1(\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0)) + \sum_{l=1}^{p-1} \left[ \frac{\partial^l}{\partial t^l} \mathbf{f}_1(\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0)) \right] \frac{\tau_k^l}{(l+1)!} \right) w_\nu, \quad (2.109)$$

kde  $\tau_k = t_{k+1} - t_k$  je velikost časového kroku. Místo použití přesného Riemannova řešiče je použit přibližný Riemannův řešič HLLC, viz kapitola 2.4.1. Vedoucí člen  $\mathbf{f}_1(\mathbf{q}^+(\tilde{\mathbf{x}}_\nu, 0))$  je tedy určen pomocí přibližného Riemannova řešiče a členy vyšších řádů jsou určeny pomocí Cauchyho-Kowalewské procedury a odpovídajících přibližných řešení Riemannova problému.

Máme-li spočten zobecněný Godunovův tok  $\hat{\mathbf{g}}_G$ , lze spočítat numerický tok  $\hat{\mathbf{H}}$ , viz (2.66), a následně pak pomocí (2.20) updatovat na numerické řešení  $\mathbf{w}_i^{k+1}$ .

Pro stabilitu schématu je nutné dodržet CFL podmínku typu (2.27) [23]:

$$\tau_k \leq \frac{\text{CFL} |D_i|}{\lambda_{i,max}^k |\partial D_i|}, \quad i \in J, \quad (2.110)$$

kde  $\text{CFL} < 1$  a:

$$\lambda_{i,max}^k = \max_{\substack{r=1,\dots,m, j \in S(i) \\ \alpha=1,\dots,\beta_{ij}, \nu=1,\dots,K_\nu}} |\lambda_r^k(\mathbf{w}_{ij,\nu}^k, \mathbf{n}_{ij}^\alpha)|, \quad m = N + 2, \quad (2.111)$$

kde  $\lambda_r^k(\mathbf{w}_{ij,\nu}^k, \mathbf{n}_{ij}^\alpha)$  jsou vlastní čísla matice  $\mathbb{A}_1$  (1.19) pro řešení zobecněného Riemannova problému ((2.38) a (2.62)) s počátečními podmínkami (2.64) v uzlových bodech  $\tilde{\mathbf{x}}_\nu$  v čase  $t = 0$ .

## 3. Adaptace výpočetní sítě

Jednou z možností zvýšení přesnosti výpočtů v případě metody konečných objemů je používat jemnější výpočetní síť. Tohoto je možno docílit tak, že buď zjemníme celou síť globálně, což vede k exponenciálnímu nárůstu počtu konečných objemů sítě s dimensí problému a nebudeme se tím v dalším textu zabývat, nebo ji výběrově zjemníme pouze v určitých oblastech. Aby se dosáhlo pokud možno co největších úspor ve výpočetní a paměťové náročnosti výpočtu, tak se mimo oblasti s velkou hodnotou normy gradientu dané veličiny též detekují oblasti s velmi malou změnou dané veličiny. A zde se pak tyto sítě zhrubí<sup>1</sup>. Metody pro adaptaci sítě lze zhruba rozdělit dle toho, zda mění strukturu výpočtové sítě, či ne, tj. přidávají, či odebírají objemy.

Primárně se zaměříme na metody neměnicí strukturu výpočtové sítě, jež dosahují adaptivity tím, že v oblastech s velkou hodnotou normy gradientu řešení zvyšují hustotu uzlů v těchto oblastech pomocí jejich přesouvání. Tyto metody se též nazývají metody relokační<sup>2</sup>. Protože je zde velmi často požadavek zachytit pohyb oblastí s velkou hodnotou normy gradientu některých počítaných veličin v čase, mluví se o tzv. *Moving Mesh*. Příkladem těchto metod jsou tzv. *variační metody*, viz kapitola 3.1, nebo metody uvedené v pracích [6, 25, 27], kde je nutné pro pohyb uzlů v čase řešit soustavu parciálních diferenciálních rovnic<sup>3</sup>. Představíme zde **námi navrženou** alternativu, jde o tzv. *VQPM* (Vertex Quality Parameter Moving Mesh Methods), kde pohybu uzlů sítě je docíleno minimalizací jisté funkce definované pro každý uzel sítě, viz kapitola (3.3).

Druhým přístupem je pomocí nějakého vhodného indikátoru, či heuristiky, rozhodnout, zda síť v daném místě zjemníme, či naopak některé konečné objemy odebereme. Existuje též možnost zkombinovat předchozí přístupy dohromady, příkladem takového přístupu je *anisotropní adaptivita*, viz kapitola (3.2).

Struktura této kapitoly je následující, nejprve je uveden popis dané adaptivní strategie. Poté následují 1D a 2D numerické experimenty zaměřené na otestování a zhodnocení dané metody adaptace sítě. Cílem těchto experimentů je ověřit chování zkoumaných adaptivních metod a vliv různých parametrů na adaptivní proces před tím, než budou aplikovány jakou součástí složitějších algoritmů.

### 3.1 Variační metody

Variační metody, viz např. práce [4, 6, 8, 27, 40], patří mezi metody relokační. V dalším předpokládejme naši fyzikální oblast  $\Omega$  a zavedme výpočetní (logickou) oblast  $\Omega_c$ . Jako  $\mathbf{x} = (x_1, \dots, x_N)$  označme souřadnice v  $\Omega$  a jako  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)$  označme souřadnice v  $\Omega_c$ , dále se zabývejme případem pro  $N = 2$ . Pro konstrukci adaptované sítě uvažujme vzájemně jednoznačné zobrazení  $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$  z  $\Omega_c$  do  $\Omega$  a inverzní zobrazení  $\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x})$  z  $\Omega$  do  $\Omega_c$ .

Naším cílem je nalézt takové vzájemně jednoznačné zobrazení z  $\Omega_c$  do  $\Omega$ , které

---

<sup>1</sup>Mesh coarsening

<sup>2</sup>Poznamenejme, že v souvislosti s metodou konečných prvků se v případě relokačních metod mluví o tzv. *r-methods* práce [5], jako variantě k metodám používajícím lokálních zjemnění a zhrubnutí sítě a změny stupně použitých polynomů tzv. *h-p methods*.

<sup>3</sup>*MMPDE* (Moving Mesh Partial Differential Equation)

rovnoměrně rozmístěným uzlům sítě v oblasti  $\Omega_c$  přiřadí uzly sítě v oblasti  $\Omega$  tak, aby se v oblastech s velkou hodnotou normy gradientu některé ze složek  $\mathbf{w}$  hustota uzlů sítě zvýšila a naopak.

V dalším se zaměříme na přístup, který využívá minimalizaci následujícího kvadratického funkcionálu:

$$E(\boldsymbol{\xi}) = \frac{1}{2} \sum_{k=1}^2 \int_{\Omega} \nabla \xi_k^T \mathbb{G}_k^{-1} \nabla \xi_k dx, \quad (3.1)$$

kde  $\mathbb{G}_k(\mathbf{x})$ ,  $k = 1, 2$  jsou symetrické pozitivně definitní matice, které budeme nazývat *monitorovací funkce*. Tyto funkce mají zásadní vliv na vlastní adaptační proces a tvar výsledné výpočetní sítě. Často se monitorovací funkce volí tak, že platí  $\mathbb{G}_k = \mathbb{G}$ ,  $k = 1, 2$ , tuto volbu použijme i v dalším textu. Popis konstrukce monitorovacích funkcí tohoto typu a jejich vliv na výslednou síť bude popsán dále v kapitole 3.1.2. Popis vlivu různých monitorovacích funkcí  $\mathbb{G}_1 = \mathbb{G}_1(\mathbf{v}_1)$  a  $\mathbb{G}_2 = \mathbb{G}_2(\mathbf{v}_2)$ , kde  $\mathbf{v}_1$  a  $\mathbf{v}_2$  jsou vektory ovlivňující směr posunu uzlů sítě, je možný nalézt např. v [4, 27].

Pro minimalizaci<sup>4</sup> výše uvedeného funkcionálu (3.1) je třeba řešit následující Eulerovy-Lagrangeovy rovnice:

$$\begin{aligned} \nabla^T \cdot (\mathbb{G}^{-1} \nabla \xi_1) &= 0, \\ \nabla^T \cdot (\mathbb{G}^{-1} \nabla \xi_2) &= 0. \end{aligned} \quad (3.2)$$

Poznamenejme, že funkcionál (3.1) vznikl jako zobecnění variační formulace tzv. *Winslowovy metody* viz kapitola 3.1.3, která používala pro adaptaci sítě speciální případ rovnic (3.2).

Protože naším cílem je konstrukce výpočetní sítě v oblasti  $\Omega$ , bude nás zajímat inverzní zobrazení  $\mathbf{x}(\boldsymbol{\xi})$  k  $\boldsymbol{\xi}(\mathbf{x})$ , které získáme záměnou závislých a nezávislých proměnných ve vztahu (3.2) a použitím transformačních vztahů:

$$\nabla \xi_1 = \frac{1}{\mathcal{J}} \begin{pmatrix} \frac{\partial x_2}{\partial \xi_2} \\ \frac{\partial \xi_2}{\partial x_1} \end{pmatrix}, \quad \nabla \xi_2 = \frac{1}{\mathcal{J}} \begin{pmatrix} \frac{-\partial x_2}{\partial \xi_1} \\ \frac{\partial x_1}{\partial \xi_1} \end{pmatrix}, \quad (3.3)$$

kde  $\mathcal{J} = \frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_1}$  je Jacobián transformace souřadnic, u kterého předpokládáme  $\mathcal{J} \neq 0$ . Aplikací výše uvedeného postupu dostáváme novou soustavu rovnic, viz např. práce [5]:

$$\begin{aligned} \frac{\partial}{\partial \xi_1} \left( \frac{\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_2}}{\mathcal{J} g} \right) - \frac{\partial}{\partial \xi_2} \left( \frac{\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_2}}{\mathcal{J} g} \right) &= 0, \\ -\frac{\partial}{\partial \xi_1} \left( \frac{\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_1}}{\mathcal{J} g} \right) + \frac{\partial}{\partial \xi_2} \left( \frac{\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_1}}{\mathcal{J} g} \right) &= 0, \end{aligned} \quad (3.4)$$

kde  $g = \det(\mathbb{G})$ . Pro vlastní adaptaci výpočtové sítě se používají rovnice (3.4)

<sup>4</sup>Existuje i možnost daný funkcionál diskretizovat a pak minimalizovat přímo, ale tato možnost se téměř nepoužívá [26].

upravené do tvaru:

$$\begin{aligned} & \left[ \frac{\partial}{\partial \xi_1} \left( \frac{\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_2}}{\mathcal{J}g} \right) - \frac{\partial}{\partial \xi_2} \left( \frac{\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_2}}{\mathcal{J}g} \right) \right] \mathbf{x}_{\xi_1} \\ & + \left[ -\frac{\partial}{\partial \xi_1} \left( \frac{\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_1}}{\mathcal{J}g} \right) + \frac{\partial}{\partial \xi_2} \left( \frac{\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_1}}{\mathcal{J}g} \right) \right] \mathbf{x}_{\xi_2} = 0. \end{aligned} \quad (3.5)$$

Numerické řešení této rovnice včetně volby příslušných okrajových podmínek bude ukázáno později v kapitole 3.1.5.

### 3.1.1 Ekvidistribuce sítě

Pokusíme se vysvětlit, proč při minimalizaci funkcionálu (3.1) dochází k adaptaci výpočtové sítě. Nejprve se zaměříme na 1D případ.

Předpokládejme, že naše fyzikální oblast  $\Omega$  je tvořena intervalem  $[a, b]$ , kde  $a < b$ . Mějme na tomto intervalu definovanou spojitou funkci  $\phi(x) > 0$ ,  $x \in [a, b]$ . Snažme se nyní rozdělit  $\Omega$  na  $J$  různých subintervalů, které jsou dány body  $a = x_1 < x_2 < \dots < x_{J+1} = b$ , tak, aby funkce  $\phi$  byla na těchto subintervalech *ekvidistribuována*, tedy aby platilo:

$$\int_{x_1}^{x_2} \phi(x) dx = \dots = \int_{x_J}^{x_{J+1}} \phi(x) dx. \quad (3.6)$$

Sít, která pro danou funkci  $\phi$  splňuje výše uvedenou podmínku (3.6), budeme nazývat *sítí ekvidistribující funkci  $\phi$* . Požadavek ekvidistribuce lze též zapsat ve tvaru:

$$\int_a^{x_i} \phi(x) dx = \frac{i-1}{J} \sigma, \quad i = 1, \dots, J+1, \quad (3.7)$$

kde

$$\sigma = \int_a^b \phi(x) dx. \quad (3.8)$$

Levou stranu vztahu (3.7) lze nahlížet jako funkci proměnné  $x_i$ , která je rostoucí, čili existuje inverzní funkce, která každému  $i = 1, \dots, J+1$  jednoznačně přiřadí pozici uzlu  $x_i$ .

Uvažujme nyní zobrazení  $x = x(\xi) : [0, 1] \rightarrow [a, b]$ , které nám přiřadí uzlové body:

$$x_i = x(\xi_i), \quad i = 1, \dots, J+1, \quad (3.9)$$

kde

$$\xi_i = \frac{i-1}{J}, \quad i = 1, \dots, J+1, \quad (3.10)$$

jsou rovnoměrně rozmístěné uzlové body v intervalu  $[0, 1]$ , jenž ztotožníme s výpočetní oblastí  $\Omega_c$ . Vztah (3.7) popisující princip ekvidistribuce přejde na:

$$\int_a^{x(\xi_i)} \phi(x) dx = \sigma \xi_i, \quad i = 1, \dots, J+1, \quad (3.11)$$

což nás vede na následující zobecnění. Spojité zobrazení  $x = x(\xi)$  splňující pro pevně dané  $\phi(x) > 0$ ,  $x \in [a, b]$  podmínku:

$$\int_a^{x(\xi)} \phi(x) dx = \sigma \xi, \quad (3.12)$$

kde  $\sigma$  je dáno pomocí vztahu (3.8), nazveme *ekvidistribuující souřadnicovou transformací*. Derivováním (3.12) dle  $\xi$  dostáváme diferenciální formulaci takové souřadnicové transformace:

$$\phi(x) \frac{dx}{d\xi} = \sigma, \quad (3.13)$$

respektive k ní inverzní transformace  $\xi = \xi(x) : [a, b] \rightarrow [0, 1]$ :

$$\frac{1}{\phi(x)} \frac{d\xi}{dx} = \frac{1}{\sigma}. \quad (3.14)$$

Všimněme si, že derivováním obou stran rovnice (3.14) dle  $x$ , dostáváme:

$$\frac{d}{dx} \left( \frac{1}{\phi(x)} \frac{d\xi}{dx} \right) = 0, \quad (3.15)$$

což odpovídá 1D případu rovnice (3.2), kde je místo monitorovací funkce  $\mathbb{G}$  v 1D použita funkce  $\phi$ , tedy požadavek ekvidistribuce vede na rovnici (3.2).

Předpokládejme, že na síti  $\mathcal{D}_h$  určené uzly  $x_i$ ,  $i = 1, \dots, J + 1$ , budeme interpolovat danou funkci  $u$ , která může být jednou ze složek řešení  $\mathbf{w}$  rovnice (1.17) v 1D. Dále předpokládejme, že ukazatel chyby, které se při této interpolaci dopustíme, lze vyjádřit ve tvaru uvedeném v práci [26]:

$$E(\mathcal{D}_h) = J^s \sum_{i=1}^J (h_i f_i)^{s+1}, \quad (3.16)$$

kde  $s > 0$  je reálné číslo a  $h_i = x_{i+1} - x_i$  představuje délku jednotlivých subintervalů  $I_i = (x_i, x_{i+1})$ . Pomocí  $f_i$  je označena aproximace střední hodnoty funkce  $f(x)$  na  $i$ -tém subintervalu, přičemž  $f(x) > 0$ ,  $x \in [a, b]$  je funkce, která obecně závisí na derivacích interpolované funkce  $u$ . Činitel  $J$  je zde přidán *ad-hoc* z důvodu asymptotické nezávislosti ukazatele  $E(\mathcal{D}_h)$  na  $J$ , tedy aby při limitním přechodu  $\max_{i=1, \dots, J} h_i \rightarrow 0$  byl tento ukazatel nenulový.

Příklad ukazatele chyby interpolace, odpovídající výše uvedenému, lze získat následovně. Uvažujme Sobolevův prostor  $H^{k+1}(a, b)$ ,  $k > 0$  a interpolujme funkci  $u \in H^{k+1}(a, b)$  pomocí po částech polynomiální funkce  $\Pi_k u$  stupně nejvýše  $k$  na síti  $\mathcal{D}_h$ , kde  $\Pi_k$  je interpolační operátor. Chyba interpolace pro  $0 \leq m \leq k$  splňuje následující omezení, viz práce [26]:

$$|u - \Pi_k u|_{H^m(a,b)}^2 \leq C \sum_{i=1}^J h_i^{1+2(k-m+1)} \langle u \rangle_{H^{k+1}(I_i)}^2, \quad (3.17)$$

kde  $|\cdot|_{H^m(a,b)}$  je *seminorma* v tomto prostoru (definici lze nalézt např. ve výše uvedené práci [26][Appendix A]),  $C$  je konstanta nezávislá na síti  $\mathcal{D}_h$  a

$$\langle u \rangle_{H^{k+1}(I_i)} = \left( \frac{1}{h_i} \int_{x_i}^{x_{i+1}} |u^{(k+1)}|^2 dx \right)^{\frac{1}{2}}, \quad (3.18)$$

kde  $u^{(k+1)}$  značí  $k + 1$  derivaci  $u$ . Toto lze přepsat do tvaru:

$$|u - \Pi_k u|_{H^m(a,b)}^2 \leq C \sum_{i=1}^J \left( h_i \langle u \rangle_{H^{k+1}(I_i)}^{\frac{2}{1+2(k-m+1)}} \right)^{1+2(k-m+1)}, \quad (3.19)$$



a regularizací pomocí konstanty  $\beta > 0$  získáme následující omezení:

$$|u - \Pi_k u|_{H^m(a,b)}^2 \leq C\beta \sum_{i=1}^J \left( h_i \left[ 1 + \frac{1}{\beta} \langle u \rangle_{H^{k+1}(I_i)}^2 \right]^{\frac{1}{1+2(k-m+1)}} \right)^{1+2(k-m+1)}. \quad (3.20)$$

Nyní lze zavést ukazatel chyby interpolace  $E(\mathcal{D}_h)$  ve tvaru (3.16) následovně:

$$E(\mathcal{D}_h) = J^{2(k-m+1)} \beta \sum_{i=1}^J \left( h_i \left[ 1 + \frac{1}{\beta} \langle u \rangle_{H^{k+1}(I_i)}^2 \right]^{\frac{1}{1+2(k-m+1)}} \right)^{1+2(k-m+1)}. \quad (3.21)$$

O vztahu ekvidistribuce a chybě, které se při výše uvedené interpolaci dopustíme, hovoří následující věta [26, Theorem 2.1.2], důkaz tamtéž.

**Věta 3.1.** *Předpokládejme, že ukazatel chyby, které se při interpolaci dopustíme, je ve tvaru (3.16), kde  $s > 0$  a  $f(x) > 0$ ,  $x \in [a, b]$  je spojitá funkce, pak platí:*

$$E(\tilde{\mathcal{D}}_h) = J^s \sum_{i=1}^J \left( \tilde{h}_i \tilde{f}_i \right)^{s+1} \geq \tilde{\sigma}_h^{s+1}, \quad \forall \tilde{\mathcal{D}}_h \in \mathcal{D}_J \quad (3.22)$$

a

$$\tilde{\sigma}_h^{s+1} \longrightarrow \sigma^{s+1}, \quad \forall \tilde{\mathcal{D}}_h \in \mathcal{D}_J,$$

kde  $\mathcal{D}_J = \left\{ \tilde{\mathcal{D}}_h : a = \tilde{x}_1 < \tilde{x}_2 < \dots < \tilde{x}_{J+1} = b \right\}$  je množina všech dělení intervalu  $[a, b]$  na  $J$  subintervalů,  $\tilde{h}_i = \tilde{x}_{i+1} - \tilde{x}_i$ ,  $\tilde{\sigma}_h = \sum_{i=1}^J \tilde{h}_i \tilde{f}_i$ ,  $\tilde{f}_i$  značí aproximaci integrálního průměru funkce  $f$  na  $i$ -tém subintervalu a  $\sigma = \int_a^b f(x) dx$ . Limitním přechodem rozumíme  $\max_{i=1, \dots, J} \tilde{h}_i \longrightarrow 0$ .

Pro síť  $\mathcal{D}_h$  ekvidistribující funkci  $f$  a splňující podmínku:

$$h_i f_i = \frac{\sigma_h}{J}, \quad i = 1, \dots, J, \quad (3.23)$$

kde  $\sigma_h = \sum_{i=1}^J h_i f_i$ , platí:

$$E(\mathcal{D}_h) = \sigma_h^{s+1} \quad (3.24)$$

a

$$\sigma_h^{s+1} \longrightarrow \sigma^{s+1} \text{ když } \max_{i=1, \dots, J} h_i \longrightarrow 0.$$

Poznamenejme, že může nastat situace, že  $\sigma_h$  pro ekvidistribující síť  $\mathcal{D}_h \in \mathcal{D}_J$  může být větší než  $\tilde{\sigma}_h$  pro síť  $\tilde{\mathcal{D}}_h \in \mathcal{D}_J$  při daném počtu subintervalů  $J$ , tedy ekvidistribující síť nemusí být sítí optimální ve smyslu odhadu  $E(\mathcal{D}_h)$ . Pouze v případě limitního přechodu  $\max_{i=1, \dots, J} h_i \longrightarrow 0$ , je ekvidistribující síť sítí optimální.

Z výše zmíněného je vidět, že v 1D vede minimalizace funkcionálu typu (3.1) na princip ekvidistribuce monitorovací funkce po výpočtové síti, kde monitorovací funkci používáme jakožto jistý ukazatel chyby interpolace. Pro případ 2D je provedena kvalitativní analýza vlivu monitorovacích funkcí na adaptivní proces v následující kapitole.

### 3.1.2 Monitorovací funkce

Pro pochopení principu vlivu monitorovacích funkcí na adaptivitu sítí uvažujme nejprve eliptický diferenciální operátor [6, 26]:

$$L = \sum_{i,j=1}^2 a_{ij}(\mathbf{x}) \frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^2 b_i(\mathbf{x}) \frac{\partial}{\partial x_i}, \quad (3.25)$$

kde  $a_{ij}(\mathbf{x})$  a  $b_i(\mathbf{x})$  jsou spojité funkce  $\mathbf{x}$  v  $\Omega$ , takové že matice  $\mathbb{A}$ :

$$\mathbb{A}(\mathbf{x}) = \begin{pmatrix} a_{11}(\mathbf{x}) & a_{12}(\mathbf{x}) \\ a_{21}(\mathbf{x}) & a_{22}(\mathbf{x}) \end{pmatrix}, \quad (3.26)$$

je symetrická pozitivně definitní pro všechna  $\mathbf{x} \in \Omega$ . Za předpokladu dostatečné hladkosti hranice  $\partial\Omega$  existuje Greenova funkce  $\mathcal{G}(\mathbf{x}, \mathbf{y})$ , která je pozitivní v  $\Omega$  a nulová na  $\partial\Omega$ . Dále uvažujme následující Dirichletův problém:

$$\begin{aligned} L[u] &= f(\mathbf{x}, u), \text{ v } \Omega, \\ u &= h(\mathbf{x}), \text{ na } \partial\Omega. \end{aligned} \quad (3.27)$$

Při znalosti Greenovy funkce  $\mathcal{G}$  můžeme psát řešení problému (3.27) ve tvaru:

$$u(\mathbf{x}) = - \int_{\Omega} \mathcal{G}(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, u(\mathbf{y})) d\mathbf{y} - \int_{\mathbf{y} \in \partial\Omega} h(\mathbf{y}) \frac{\partial \mathcal{G}}{\partial \mathbf{n}}(\mathbf{x}, \mathbf{y}) dS, \quad (3.28)$$

kde  $\mathbf{n}$  je vnější normála k hranici  $\partial\Omega$  a  $\frac{\partial \mathcal{G}}{\partial \mathbf{n}}$  je derivace Greenovy funkce ve směru vnější normály. Budeme-li uvažovat problém typu (3.27) s nulovou pravou stranou:

$$\begin{aligned} L[v] &= 0, \text{ v } \Omega, \\ v &= h(\mathbf{x}), \text{ na } \partial\Omega, \end{aligned} \quad (3.29)$$

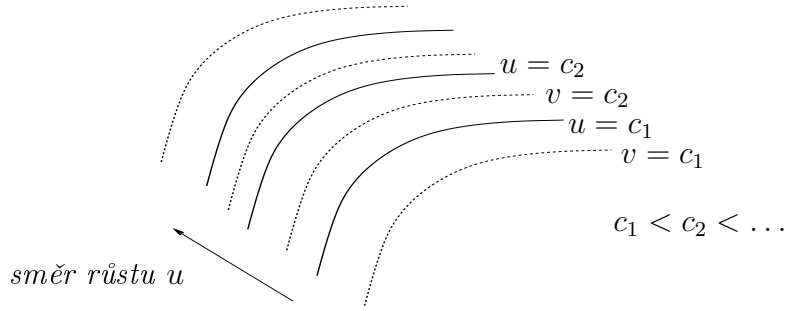
lze pro jeho řešení  $v$  psát:

$$v(\mathbf{x}) = - \int_{\mathbf{y} \in \partial\Omega} h(\mathbf{y}) \frac{\partial \mathcal{G}}{\partial \mathbf{n}}(\mathbf{x}, \mathbf{y}) dS. \quad (3.30)$$

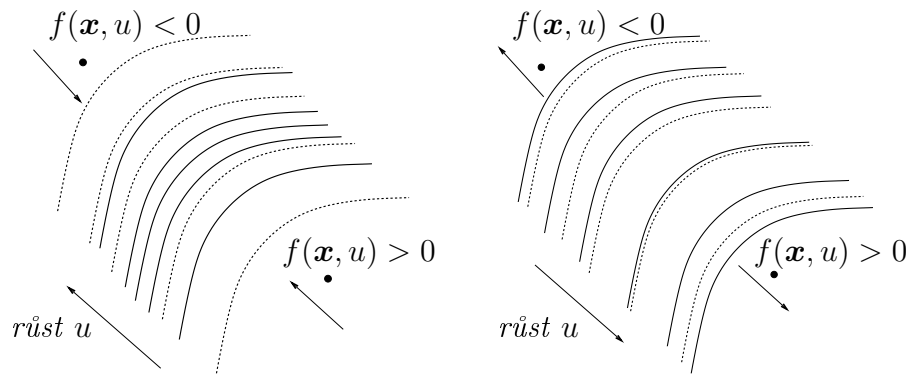
A tedy řešení  $u(\mathbf{x})$  původního Dirichletova problému lze zapsat pomocí řešení  $v(\mathbf{x})$  "homogenní" úlohy (3.29) následovně:

$$u(\mathbf{x}) = v(\mathbf{x}) - \int_{\Omega} \mathcal{G}(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, u(\mathbf{y})) d\mathbf{y}. \quad (3.31)$$

V dalším se zaměříme na geometrickou interpretaci. Uvažujme nejprve, že je  $f(\mathbf{x}, u) > 0$ , tak díky pozitivnosti  $G(\mathbf{x}, \mathbf{y})$  leží křivka  $u(\mathbf{x}) = c$  v oblasti, kde  $v(\mathbf{x}) \geq c$ . Dochází tedy k posunu izokřivek řešení  $u(\mathbf{x})$  vůči řešení  $v(\mathbf{x})$  ve směru růstu  $u(\mathbf{x})$ , viz obrázek 3.1, a naopak při  $f(\mathbf{x}, u) < 0$ ,  $\mathbf{x} \in \Omega$  dojde k posunu opačným směrem. Protože Greenova funkce se chová singularně [26], tak v integrálu ve vztahu (3.31) se projeví převážně body z jistého okolí bodu  $\mathbf{x}$  a v případě kladné a dostatečně vysoké hodnoty funkce  $f(\mathbf{x}, u)$  na tomto okolí zde dochází k výše zmíněnému posunu izokřivek řešení ve směru růstu  $u(\mathbf{x})$ . A obráceně, v



Obrázek 3.1: Vzájemné posunutí izokřivek řešení nehomogenní a homogenní elip-  
tické úlohy.



Obrázek 3.2: Komprese izokřivek (izokřivky  $u$  plná čára,  $v$  čárkova-  
né) Obrázek 3.3: Expanze izokřivek (izokřivky  $u$  plná čára,  $v$  čárkova-  
né)

případě záporné hodnoty funkce  $f(\mathbf{x}, u)$  na tomto okolí dochází k posunu izokřiv-  
vek řešení proti směru růstu  $u(\mathbf{x})$ . Toto chování vede k tomu, že v případě změny  
znaménka funkce  $f(\mathbf{x}, u)$  mezi dvěma body zde dochází ke kompresi, či expanzi  
izokřivek  $u(\mathbf{x})$  vůči řešení  $v(\mathbf{x})$  homogenní úlohy, viz obrázky 3.2 a 3.3.

Protože monitorovací funkce  $\mathbb{G}$ , resp. její inverze  $\mathbb{G}^{-1}$  jsou symetrické pozitivně  
definitní matice, lze je zapsat ve tvaru:

$$\mathbb{G} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T, \quad (3.32)$$

$$\mathbb{G}^{-1} = \frac{1}{\lambda_1} \mathbf{v}_1 \mathbf{v}_1^T + \frac{1}{\lambda_2} \mathbf{v}_2 \mathbf{v}_2^T, \quad (3.33)$$

kde  $\lambda_1, \lambda_2 > 0$  jsou vlastní čísla  $\mathbb{G}$  a  $\mathbf{v}_1$  a  $\mathbf{v}_2$  jsou odpovídající ortogonální nor-  
malizované vlastní vektory. Zavedením derivace ve směru  $\mathbf{v}$ :

$$\frac{\partial}{\partial \mathbf{v}} = \mathbf{v}^T \nabla, \quad (3.34)$$

a použitím (3.32) lze funkcionál (3.1) přepsat do tvaru:

$$E(\boldsymbol{\xi}) = \frac{1}{2} \int_{\Omega} \left[ \frac{1}{\lambda_1} \left( \left| \frac{\partial \xi_1}{\partial \mathbf{v}_1} \right|^2 + \left| \frac{\partial \xi_2}{\partial \mathbf{v}_1} \right|^2 \right) + \frac{1}{\lambda_2} \left( \left| \frac{\partial \xi_1}{\partial \mathbf{v}_2} \right|^2 + \left| \frac{\partial \xi_2}{\partial \mathbf{v}_2} \right|^2 \right) \right] dx dy, \quad (3.35)$$

a příslušné Eulerovy-Lagrangeovy rovnice pro minimalizaci tohoto funkcionálu jsou:

$$\nabla^T \left( \frac{\mathbf{v}_1}{\lambda_1} \frac{\partial \xi_1}{\partial \mathbf{v}_1} \right) + \nabla^T \left( \frac{\mathbf{v}_2}{\lambda_2} \frac{\partial \xi_1}{\partial \mathbf{v}_2} \right) = 0, \quad (3.36)$$

$$\nabla^T \left( \frac{\mathbf{v}_1}{\lambda_1} \frac{\partial \xi_2}{\partial \mathbf{v}_1} \right) + \nabla^T \left( \frac{\mathbf{v}_2}{\lambda_2} \frac{\partial \xi_2}{\partial \mathbf{v}_2} \right) = 0. \quad (3.37)$$

Rozepsáním naznačených derivací a využitím (3.34) získáme následující rovnice:

$$\frac{\partial^2 \xi_1}{\partial \mathbf{v}_1^2} + \frac{\lambda_1}{\lambda_2} \frac{\partial^2 \xi_1}{\partial \mathbf{v}_2^2} = \left( \frac{1}{\lambda_1} \frac{\partial \lambda_1}{\partial \mathbf{v}_1} - \nabla^T \mathbf{v}_1 \right) \frac{\partial \xi_1}{\partial \mathbf{v}_1} + \frac{\lambda_1}{\lambda_2} \left( \frac{1}{\lambda_2} \frac{\partial \lambda_2}{\partial \mathbf{v}_2} - \nabla^T \mathbf{v}_2 \right) \frac{\partial \xi_1}{\partial \mathbf{v}_2} \quad (3.38)$$

$$\frac{\partial^2 \xi_2}{\partial \mathbf{v}_1^2} + \frac{\lambda_1}{\lambda_2} \frac{\partial^2 \xi_2}{\partial \mathbf{v}_2^2} = \left( \frac{1}{\lambda_1} \frac{\partial \lambda_1}{\partial \mathbf{v}_1} - \nabla^T \mathbf{v}_1 \right) \frac{\partial \xi_2}{\partial \mathbf{v}_1} + \frac{\lambda_1}{\lambda_2} \left( \frac{1}{\lambda_2} \frac{\partial \lambda_2}{\partial \mathbf{v}_2} - \nabla^T \mathbf{v}_2 \right) \frac{\partial \xi_2}{\partial \mathbf{v}_2} \quad (3.39)$$

Vydeme-li z první z předchozích rovnic a definujeme následující eliptický operátor [6]:

$$L[\xi_1] := \frac{\partial^2 \xi_1}{\partial \mathbf{v}_1^2} + \frac{\lambda_1}{\lambda_2} \frac{\partial^2 \xi_1}{\partial \mathbf{v}_2^2} + (\nabla^T \mathbf{v}_1) \frac{\partial \xi_1}{\partial \mathbf{v}_1} - \frac{\lambda_1}{\lambda_2} \left( \frac{1}{\lambda_2} \frac{\partial \lambda_2}{\partial \mathbf{v}_2} - \nabla^T \mathbf{v}_2 \right) \frac{\partial \xi_1}{\partial \mathbf{v}_2}, \quad (3.40)$$

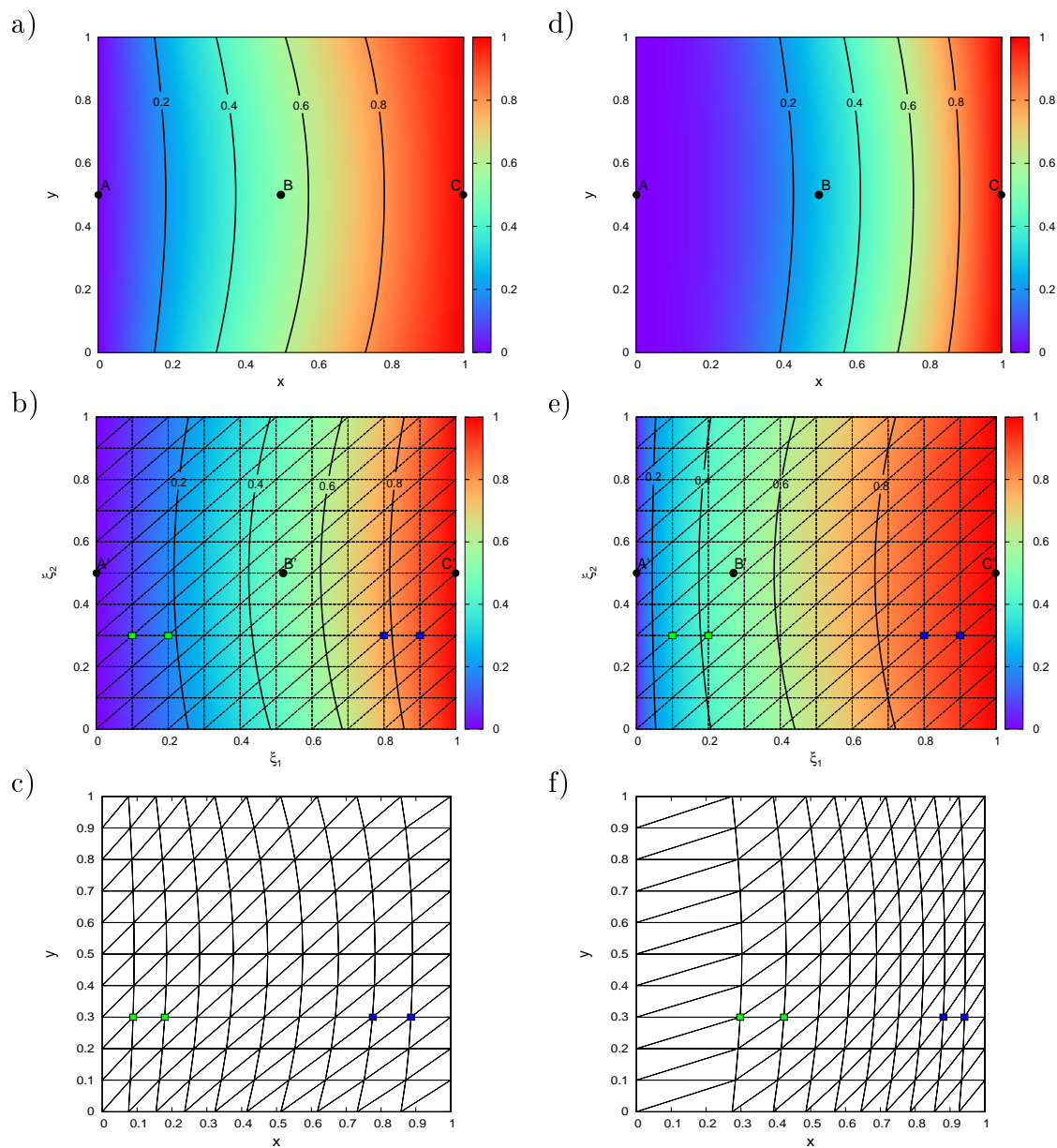
kde zdrojový člen  $f(\mathbf{x}, u)$ :

$$f(\mathbf{x}, \xi_1) := \frac{1}{\lambda_1} \frac{\partial \lambda_1}{\partial \mathbf{v}_1} \frac{\partial \xi_1}{\partial \mathbf{v}_1}, \quad (3.41)$$

vidíme, že Eulerova-Lagrangeova rovnice (3.36) odpovídá problému (3.27).

Aplikujme nyní kvalitativní analýzu zmíněnou v předchozím textu. Protože zdrojový člen  $f(\mathbf{x}, u)$  mění znaménko v závislosti na  $\frac{1}{\lambda_1} \frac{\partial \lambda_1}{\partial \mathbf{v}_1}$ , a tedy se projeví efekt expanze, či komprese izokřivek řešení v závislosti na změně  $\lambda_1$  ve směru  $\mathbf{v}_1$ , tento efekt je relativní vůči řešení problému (3.29) s operátorem (3.40). Komprese resp. expanze izokřivek řešení úlohy se na adaptivitě sítě projeví jako zjemnění, resp. zhrubnutí výpočetní sítě v daném směru. Na výsledném posunutí, se ovšem projeví i vliv dalších členů operátoru (3.40), jako je např. změna  $\lambda_2$  ve směru  $\mathbf{v}_2$ , což ve výsledku činí přesnou analýzu velmi komplikovanou. Tyto efekty, tedy efekty od druhého, třetího a čtvrtého členu, budeme ve shodě s [6] nazývat dvoudimensionální efekty a jejich vliv, vyjma vlivu třetího členu (tedy změny  $\xi_1$  ve směru  $\mathbf{v}_1$ ) lze redukovat zmenšením poměru  $\frac{\lambda_1}{\lambda_2}$ , viz následující část textu věnovaná vlivu vlastních čísel. Poznamenejme, že podobnou analýzu bychom mohli provést samozřejmě i pro druhou z rovnic (3.39).

Výše uvedené si pro názornost ilustrujeme na následujících obrázcích 3.4. Předpokládejme, že máme k dispozici zobrazení  $\xi_1(x, y)$  z  $\Omega$  do  $\Omega_c$ , jehož izokřivky jsou zaneseny na obrázku 3.4 (a) a, abychom vyloučili výše zmíněné dvoudimensionální efekty, uvažujme zobrazení  $\xi_2(x, y) = y$ . Vezmeme-li si zde libovolnou dvojici bodů, např.  $A, B$ , či  $B, C$ , a zobrazíme-li je do  $\Omega_c$ , dostáváme body  $A' = [\xi_1(A), \xi_2(A)]$ ,  $B' = [\xi_1(B), \xi_2(B)]$ ,  $C' = [\xi_1(C), \xi_2(C)]$ . Z obrázku (a), (b), resp. (d), (e) je vidět, že čím více izokřivek  $\xi_{1,i}(x, y) = ic$ ,  $i = 1, 2, \dots$ ,  $c \in \mathbb{R}$  je mezi danými dvěma body, tak tím je jejich vzdálenost z hlediska souřadnice  $\xi_1$  v  $\Omega_c$  větší. Vezmeme-li nyní dva vrcholy sítě v  $\Omega_c$  (modré, resp. zelené čtverečky na obrázcích 3.4 (b), (e), pak při zobrazení sítě  $x = x(\xi_1, \xi_2)$  a  $y = \xi_2$  z  $\Omega_c$  do  $\Omega$  (čili adaptaci sítě), dojde k jejich posunu, který je tím výraznější, čím více jsou vůči sobě posunuty izokřivky  $\xi_1$  homogenní (obrázek (a)) a nehomogenní úlohy (obrázek (d)).



Obrázek 3.4: Ilustrace vlivu komprese a expanze izokřivky. Vlevo (a) izokřivky  $\xi_1(x, y)$  v oblasti  $\Omega$ , (b) izokřivky  $x(\xi_1, \xi_2)$  v oblasti  $\Omega_c$ , (c) zobrazená (adaptovaná) síť v oblasti  $\Omega$ , pro homogenní úlohu. Vpravo (d) izokřivky  $\xi_1(x, y)$  v oblasti  $\Omega$ , (e) izokřivky  $x(\xi_1, \xi_2)$  v oblasti  $\Omega_c$ , (f) zobrazená (adaptovaná) síť v oblasti  $\Omega$ , pro nehomogenní úlohu.

## Vliv vlastních čísel $\lambda$

Zde si ilustrujeme vliv vlastních čísel  $\lambda$  monitorovací funkce  $\mathbb{G}$  a jejich směrových derivací na adaptivitu výpočetní sítě. Uvažujme pevně zvolené vektory

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

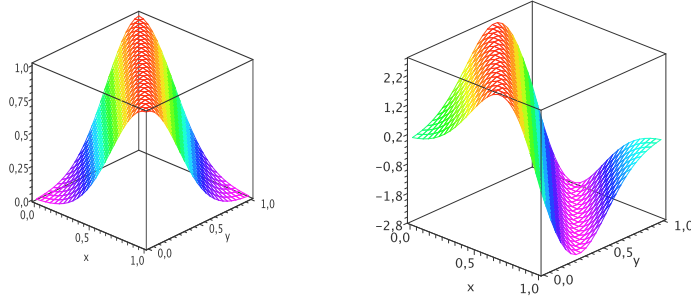
a hodnotu  $\lambda_1$  zvolme :

$$\lambda_1 = e^{-5(x+y-1)^2},$$

tedy

$$\frac{\partial \lambda_1}{\partial \mathbf{v}_1} = \sqrt{2}(-10x - 10y + 10) e^{-5(x+y-1)^2}.$$

Na obrázku 3.5 (vpravo) je vidět, že změna znaménka  $\frac{\partial \lambda_1}{\partial \mathbf{v}_1}$  odpovídá přesně té situaci, která je potřebná pro kompresi izokřivek a tedy pro zjemnění sítě v daném směru.



Obrázek 3.5: Průběh  $\lambda_1$  a  $\frac{\partial \lambda_1}{\partial \mathbf{v}_1}$  na  $[0, 1] \times [0, 1]$

Nyní sestrojme dle vztahu (3.32) monitorovací funkci  $\mathbb{G}$  tak, že budeme volit  $\lambda_2 = \alpha \lambda_1$ ,  $\alpha = 0.5, 1, 2, 4$ . Numericky vyřešíme soustavu (3.5), kde oblasti  $\Omega$  i  $\Omega_c$  jsou voleny jako čtverec  $[0, 1] \times [0, 1]$  a byla použita Dirichletova okrajová podmínka, taková, že uzly jsou na hranici pevné. Podrobnější diskuse způsobu řešení uvedené rovnice bude provedena dále. Na obrázcích 3.6 je ukázán vliv poměru  $\lambda_1/\lambda_2 = 2, 1, 0.5, 0.25$  na výslednou síť. Je vidět, že s klesajícím poměrem  $\lambda_1/\lambda_2$  je posun uzlů sítě ve směru  $\mathbf{v}_1$  výraznější.

Pro ilustraci druhého z výše zmíněných jevů, expanze izokřivek a tedy zhrubnutí dané sítě, uvažujme stejné vektory  $\mathbf{v}_1$  a  $\mathbf{v}_2$  jako v předchozím příkladu, ale použijme následující předpis  $\lambda_1$

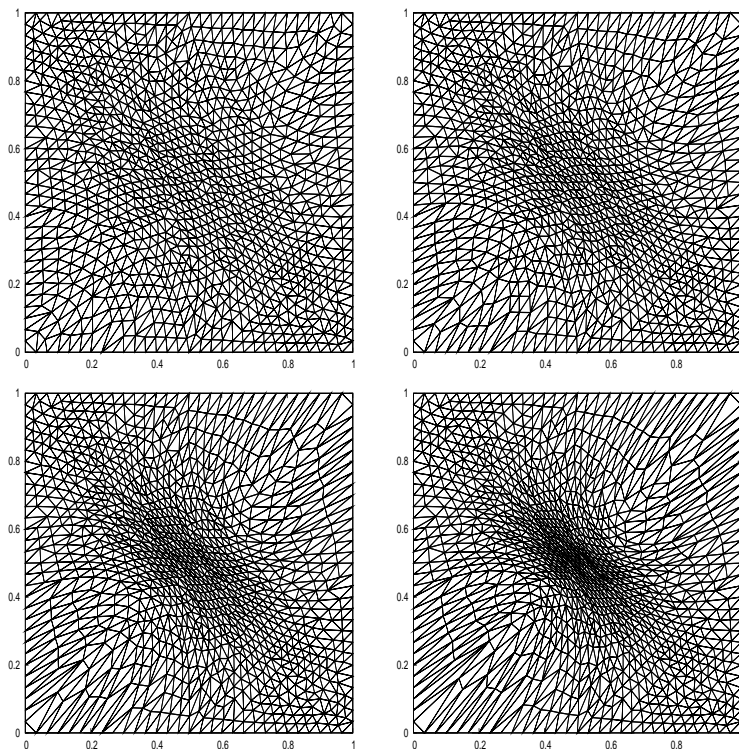
$$\lambda_1 = 2 - \operatorname{sech}(50(x+y-1)^2),$$

kde funkce hyperbolický sekans je definována následovně:

$$\operatorname{sech}(x) := \frac{2}{\exp x + \exp -x}. \quad (3.42)$$

Pro derivaci ve směru  $\mathbf{v}_1$  platí:

$$\frac{\partial \lambda_1}{\partial \mathbf{v}_1} = 100 \frac{\sqrt{2}(x+y-1) \sinh(50(x+y-1)^2)}{(\cosh(50(x+y-1)^2))^2}.$$



Obrázek 3.6: Vliv poměru  $\lambda_1/\lambda_2 = 2, 1, 0.5, 0.25$  na výslednou síť v  $\Omega$ , 2112 objemů

Průběh  $\lambda_1$  a  $\frac{\partial \lambda_1}{\partial \mathbf{v}_1}$  je znázorněn na obrázku 3.7. Na obrázku 3.8 je zachycena výsledná síť. Poznamenejme, že pro námi použitý poměr  $\lambda_1/\lambda_2 = 0.25$  je expanze mnohem méně zřetelná, proto byl použit větší počet objemů sítě než v předcházející ukázce.

### 3.1.3 Volba monitorovacích funkcí

Dále se zaměříme na možné volby monitorovacích funkcí, jednou z nejjednodušších voleb je Winslowova metoda, kde monitorovací funkci  $\mathbb{G}$  volíme ve tvaru:

$$\mathbb{G} = w(\mathbf{x}) \mathbb{I}, \quad (3.43)$$

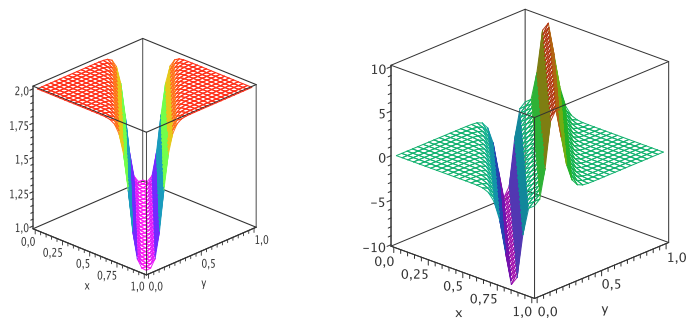
kde  $\mathbb{I}$  je jednotková matice a  $w(\mathbf{x})$  je pozitivní váhová funkce, která je volena ve tvaru:

$$w(\mathbf{x}) = \sqrt{1 + \alpha |\nabla u|^2}, \quad (3.44)$$

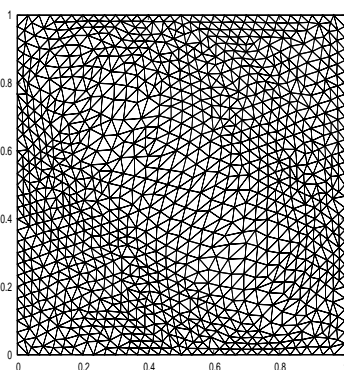
kde  $u$  je buď přímo některá ze složek stavového vektoru  $\mathbf{w}$ , nebo veličina ze stavového vektoru spočtená<sup>5</sup> a  $\alpha$  je váhový parametr, jehož hodnotou lze ovlivňovat míru zhuštění sítě poblíž oblasti s velkou hodnotou  $|\nabla u|$ . Poznamenejme, že v některých zdrojích se rozumí Winslowovou metodou použití (3.44) s  $\alpha = 1$ . Protože monitorovací funkci lze zapsat ve tvaru (3.32), je vidět, že Winslowova metoda odpovídá následující volbě vlastních čísel  $\lambda$ :

$$\lambda_1 = \lambda_2 = w(\mathbf{x}) \quad (3.45)$$

<sup>5</sup>V případě detekce kontaktní nespojitosti se používá např. v práci [33] entropie.



Obrázek 3.7: Průběh  $\lambda_1$  a  $\frac{\partial \lambda_1}{\partial \mathbf{v}_1}$  na  $[0, 1] \times [0, 1]$



Obrázek 3.8: Ukázka vlivu expanze izokřivek na výslednou síť,  $\lambda_1/\lambda_2 = 0.25$ , 3806 objemů

a  $\mathbf{v}_1$  a  $\mathbf{v}_2$  jsou libovolné dva navzájem ortonormální vektory. Adaptace sítě touto metodou je tedy izotropní v tom smyslu, že není předepsán nějaký konkrétní směr. Nejvíce se však projeví směr, ve kterém je změna  $w(\mathbf{x})$  největší.

Zobecněním Winslowovy metody dostáváme následující třídu funkcí:

$$\begin{aligned} \mathbf{v}_1 &= \frac{\nabla u}{|\nabla u|}, \mathbf{v}_2 \text{ ortonormální k } \mathbf{v}_1 \\ \lambda_1 &= \sqrt{1 + |\nabla u|^2}, \\ \lambda_2 &= f(\lambda_1), \end{aligned} \tag{3.46}$$

kde  $f(\lambda_1)$  je kladná funkce. Winslowova metoda odpovídá volbě vlastního čísla  $\lambda_2 = \lambda_1$  a  $\alpha = 1$ .

Budeme-li uvažovat matici  $\mathbb{G}$  ve tvaru:

$$\mathbb{G} = \frac{1}{\sqrt{m}} \mathbb{M}, \tag{3.47}$$

kde  $\mathbb{M}(\mathbf{x})$  je symetrická pozitivně definitní matice a  $m = \det(\mathbb{M})$ , získáme třídu monitorovacích funkcí založených na *harmonickém zobrazení*<sup>6</sup>. Označíme-li  $\alpha_1$  a

<sup>6</sup>Definice převzatá z [26]. Harmonické zobrazení  $\xi = \xi(\mathbf{x})$  z  $M$  do  $N$ , kde  $M, N$  jsou Riemannovy manifoldy, je takové zobrazení, které je extrémou funkcionálu  $E(\xi) = \int_M \sqrt{m} \sum_{i,j,\alpha,\beta} m^{ij} h_{\alpha\beta} \frac{\partial \xi_\alpha}{\partial x_i} \frac{\partial \xi_\beta}{\partial x_j} dx$ , kde  $m_{ij}$ , resp.  $h_{\alpha\beta}$  jsou metrické tenzory manifoldů  $M$ , resp.



$\alpha_2$  vlastní čísla matice  $\mathbb{M}$ , pak pro vlastní vektor  $\mathbf{v}_1$  a vlastní čísla matice  $\mathbb{G}$ ,  $\lambda_1$  a  $\lambda_2$  platí:

$$\begin{aligned} \mathbf{v}_1 \text{ je vlastní vektor k } \alpha_1, \\ \lambda_1 = \sqrt{\frac{\alpha_1}{\alpha_2}}, \\ \lambda_2 = \sqrt{\frac{\alpha_2}{\alpha_1}}, \end{aligned} \tag{3.48}$$

Výhodou metod tohoto typu je, že pro 2D problémy, kde  $\Omega_c$  je konvexní a zobrazení z  $\partial\Omega$  do  $\partial\Omega_c$  je hladké, je zaručena existence a jednoznačnost zobrazení mezi  $\Omega$  a  $\Omega_c$ , viz např. [5, 4, 7]. Příkladem takovéto funkce je funkce<sup>7</sup>, kde volíme:

$$\mathbb{M} = \mathbb{I} + \nabla u \nabla u^T, \tag{3.49}$$

a tedy  $\alpha_1 = 1 + |\nabla u|^2$ ,  $\alpha_2 = 1$  a  $\sqrt{m} = \sqrt{1 + |\nabla u|^2}$ .

Dále si všimněme, že spočteme-li hodnoty  $\lambda$  pomocí (3.48) tedy,  $\lambda_1 = \sqrt{m}$  a  $\lambda_2 = 1/\sqrt{m}$  a  $\mathbf{v}_1 = \nabla u / |\nabla u|$ , pak vidíme, že tato funkce je speciálním případem monitorovací funkce (3.46) pro volbu  $\lambda_2 = 1/\lambda_1$ . Poznamenejme, že v případě jedné dimenze, odpovídá funkce (3.49) monitorovací funkci (3.44) pro Winslowovu metodu s volbou  $\alpha = 1$ .

Diskuse volby vlivu jednotlivých monitorovacích funkcí a jejich parametrů bude provedena v části práce zabývající se numerickými experimenty za použití těchto metod.

### 3.1.4 Numerické řešení MMPDE v 1D

V této části se zaměříme na možnosti numerického řešení rovnic pro pohyb sítě v jedné dimenzi. Pro tento případ lze rovnice (3.2), resp. (3.5) zapsat následovně:

$$(G^{-1}(x)\xi_x)_x = 0, \tag{3.50}$$

resp.

$$(G(x)x_\xi)_\xi = 0, \tag{3.51}$$

kde  $G$  je monitorovací funkce v 1D, v dalším uvažujme monitorovací funkci typu (3.43), tedy Winslowovu metodu v 1D. Dále předpokládejme, že fyzikální oblast  $\Omega$  je tvořena intervalem  $[a, b]$ , kde  $a < b$  a výpočetní oblast  $\Omega_c$  je interval  $[0, 1]$ . Naším cílem v této fázi je numericky řešit rovnici (3.51) za podmínek:

$$\begin{aligned} x(0) &= a, \\ x(1) &= b. \end{aligned} \tag{3.52}$$

Jeden ze způsobů jak rovnici (3.51) řešit, je použití *metody sítě*. Použití této metody vede na řešení systému nelineárních rovnic [29], jehož řešení (i numerické) je časově značně náročné, obzvláště v situaci, kdy adaptace sítě je jen jednou

---

$N$ , a  $\mathbb{M} = (m_{ij})$ ,  $m = \det(\mathbb{M})$ ,  $\mathbb{M}^{-1} = (m^{ij})$ . Všimněme si, že ztotožníme-li  $\Omega$ , resp.  $\Omega_c$  s  $M$ , resp. s  $N$ , tak při volbě tenzoru  $h_{\alpha\beta} = \delta_{\alpha\beta}$ , kde  $\delta_{\alpha\beta}$  je Kroneckerovo delta, přechází výše uvedený funkcionál na funkcionál (3.1) s monitorovací funkcí  $\mathbb{G}$  ve tvaru (3.47).

<sup>7</sup>Funkce tohoto typu jsou nazývány *arc-length like monitor function*. Protože tuto funkci budeme často používat, označme ji zkratkou ALLMF.

z rutin celkového algoritmu. Tato nelineárnost je způsobena tím, že námi uvažovaná monitorovací funkce  $G$ , viz (3.43), závisí jak na  $x$ , tak i na  $\frac{du}{dx}$ . Navíc v numerických experimentech provedených v práci [26] se ukazuje, že tato metoda ne vždy konverguje v případě malého počtu (desítky) uzlů sítě.

Další ze způsobů jak rovnici (3.51) řešit, je použít *metodu umělého času*. Zavedením umělého času  $\tau$  se převede řešení rovnice (3.51) na nalezení stacionárního stavu pro následující problém:

Najít funkci  $x = x(\xi, \tau)$  takovou, že

$$x_\tau = (G(x)x_\xi)_\xi, \quad 0 < \xi < 1, \quad \tau > 0, \quad (3.53)$$

s okrajovou podmínkou  $x(0, \tau) = a$  a  $x(1, \tau) = b$  pro  $\tau > 0$  a počáteční podmínkou odpovídající rovnoměrné ekvidistribuci na intervalu  $[a, b]$ , tedy

$$x(\xi, 0) = (b - a)\xi + a, \quad \text{kde } \xi \in [0, 1].$$

Zavedme časovou diskretizaci  $\tau_\nu = \nu \Delta\tau$ ,  $\nu = 0, 1, 2, \dots$ , kde  $\Delta\tau$  je velikost časového kroku a předpokládejme, že máme 1D síť konečných objemů  $\mathcal{D}_h$ ,  $\mathcal{D}_h = \{D_i\}_{i=1, \dots, J}$ , kde uzly  $x_i$  a  $x_{i+1}$  tvoří objem  $D_i$ ,  $J$  je počet objemů. Označme střed objemu  $D_i$ , jako  $x_{i+\frac{1}{2}} = (x_{i+1} + x_i)/2$ . Aproximaci veličiny  $u(x, \tau)$  v bodě  $x_{i+\frac{1}{2}}$  v čase  $\tau_\nu$  označme jako  $u_{i+\frac{1}{2}}^\nu$ . Použijeme-li přímo  $j$ -tou složku konečně objemového řešení  $\mathbf{w}_i^k$  na objemu  $D_i$ , pak platí  $u_{i+\frac{1}{2}}^0 = (\mathbf{w}_i^k)_j$ . Diskretizací rovnice (3.53) pomocí metody sítí a jednoduchou úpravou obdržíme toto výpočetní schéma:

$$x_i^{\nu+1} = x_i^\nu + \frac{\Delta\tau}{h^2} \left( G(u_{i+\frac{1}{2}}^\nu)(x_{i+1}^\nu - x_i^\nu) - G(u_{i-\frac{1}{2}}^\nu)(x_i^\nu - x_{i-1}^\nu) \right), \quad i = 2, \dots, J, \quad (3.54)$$

kde  $G(u_{i\pm\frac{1}{2}}^\nu)$  značí hodnotu monitorovací funkce (3.43) zkonstruované pomocí  $u_{i\pm\frac{1}{2}}^\nu$ ,  $h = 1/J$  je velikost prostorového kroku na  $\Omega_c$ .

Pro výpočet derivace  $u_{i\pm\frac{1}{2}}^\nu$  v  $G$  používáme v naší práci centrální diference. Horní index  $\nu$  je u veličiny  $u_{i\pm\frac{1}{2}}^\nu$  z toho důvodu, že změna objemu  $D_i$  v časovém kroku  $\nu$  je spojena se změnou hodnoty řešení k tomuto objemu příslušející a tedy i s hodnotou  $u_{i+\frac{1}{2}}^\nu$ , další detaily k přepočtu hodnot lze nalézt v kapitole 3.1.6 a kapitole 4.2.1. Při praktické realizaci použijeme, dle [40, 25], místo monitorovací funkce  $G(u_{i+\frac{1}{2}}^\nu)$  v (3.54) následující zhlazení:

$$\tilde{G}(u_{i+\frac{1}{2}}^\nu) = \frac{1}{4} \left( G(u_{i-\frac{1}{2}}^\nu) + 2G(u_{i+\frac{1}{2}}^\nu) + G(u_{i+\frac{3}{2}}^\nu) \right), \quad (3.55)$$

a místo hodnoty  $G(u_{i-\frac{1}{2}}^\nu)$  použijeme  $\tilde{G}(u_{i-\frac{1}{2}}^\nu)$  určenou analogicky s tím, že pro krajní objemy  $D_i$ , kde nelze použít tohoto zhlazení, používáme přímo hodnotu  $G(u_{i\pm\frac{1}{2}}^\nu)$ . Hodnoty krajních uzlů jsou dány pomocí okrajové podmínky, tedy  $x_1 = a$  a  $x_{J+1} = b$ . Při výpočtu pomocí předcházejícího schématu (3.54) je nutné splnit následující podmínku stability:

$$\max_i \left( \frac{\Delta\tau}{h^2} G(u_{i+\frac{1}{2}}^\nu) \right) \leq \frac{1}{2}. \quad (3.56)$$

V dalším postupujeme dle [40]. Přepíšeme-li si rovnici (3.54) do tvaru:

$$x_i^{\nu+1} = x_i^\nu + \alpha_{i+\frac{1}{2}}(x_{i+1}^\nu - x_i^\nu) + \alpha_{i-\frac{1}{2}}(x_{i-1}^\nu - x_i^\nu), \quad (3.57)$$

kde  $\alpha_{i\pm\frac{1}{2}} = \frac{\Delta\tau}{h^2}G(u_{i\pm\frac{1}{2}}^\nu)$ , tak, když platí  $x_i^\nu < x_{i+1}^\nu$   $i = 1, \dots, J$  a je splněna podmínka stability, pak:

$$x_i^{\nu+1} \in (x_{i-1}^\nu, x_{i+1}^\nu) \quad i = 2, \dots, J.$$

Dále, odečteme-li od rovnice (3.57) rovnici:

$$x_{i-1}^{\nu+1} = x_{i-1}^\nu + \alpha_{i-\frac{1}{2}}(x_i^\nu - x_{i-1}^\nu) + \alpha_{i-\frac{3}{2}}(x_{i-2}^\nu - x_{i-1}^\nu),$$

dostáváme:

$$\begin{aligned} x_i^{\nu+1} - x_{i-1}^{\nu+1} &= \alpha_{i+\frac{1}{2}}(x_{i+1}^\nu - x_i^\nu) + (1 - 2\alpha_{i-\frac{1}{2}})(x_i^\nu - x_{i-1}^\nu) + \\ &+ \alpha_{i-\frac{3}{2}}(x_{i-1}^\nu - x_{i-2}^\nu), \end{aligned}$$

a tedy, jsou-li splněny výše uvedené požadavky, je pravá strana tohoto výrazu kladná, což vede k tomu, že  $x_i^{\nu+1} > x_{i-1}^{\nu+1}$   $i = 2, \dots, J+1$ . Vidíme tedy, že podmínka (3.56) zaručuje, že nedojde k destrukci (přesmýknutí uzlů) sítě.

Splnění podmínky stability je však omezující, protože např. pro výpočetní oblast s vysokým počtem objemů, či pro takovou konfiguraci řešení, které odpovídá velká hodnota  $G(u_{i+\frac{1}{2}}^\nu)$ , je nutné volit velmi malý časový krok  $\tau$ , což značně zpomaluje výpočet. Uvažujme diskretizaci, kterou získáme z (3.54) použitím v  $\nu$ -tém časovém kroku hodnot  $x_{i-1}^{\nu+1}$  a  $x_i^{\nu+1}$  místo  $x_{i-1}^\nu$  a  $x_i^\nu$ , aplikací výše uvedeného dostáváme následující schéma:

$$x_i^{\nu+1} = \frac{G(u_{i+\frac{1}{2}}^\nu)x_{i+1}^\nu + G(u_{i-\frac{1}{2}}^\nu)x_{i-1}^{\nu+1}}{G(u_{i+\frac{1}{2}}^\nu) + G(u_{i-\frac{1}{2}}^\nu)}. \quad (3.58)$$

Tuto metodu budeme díky použité diskretizaci ve shodě s prací [40] nazývat Gaussovým-Seidelovým iteračním schématem (G-S iterace). V dalším postupujeme dle výše uvedené práce [40]. Zavedením  $p_i = \frac{G(u_{i+\frac{1}{2}}^\nu)}{G(u_{i+\frac{1}{2}}^\nu) + G(u_{i-\frac{1}{2}}^\nu)}$ ,  $q_i = \frac{G(u_{i-\frac{1}{2}}^\nu)}{G(u_{i+\frac{1}{2}}^\nu) + G(u_{i-\frac{1}{2}}^\nu)}$  lze převést schéma (3.58) do tvaru:

$$-p_i x_{i+1}^\nu + x_i^{\nu+1} - q_i x_{i-1}^{\nu+1} = 0,$$

a dále využitím, že  $p_i + q_i = 1$  a po úpravách obdržíme následující vztah:

$$x_i^{\nu+1} - x_{i+1}^\nu - q_i (x_{i-1}^{\nu+1} - x_i^\nu) = q_i (x_i^\nu - x_{i+1}^\nu).$$

Z předpokladu  $x_i^\nu < x_{i+1}^\nu$  a nezápornosti monitorovacích funkcí ( $p_i, q_i > 0$ ) plyne, že pravá strana předchozího výrazu je záporná a tedy:

$$x_i^{\nu+1} - x_{i+1}^\nu < q_i (x_{i-1}^{\nu+1} - x_i^\nu),$$

což lze pomocí rekurze zapsat:

$$x_i^{\nu+1} - x_{i+1}^\nu < \left( \prod_{k=2}^i q_k \right) (x_1^{\nu+1} - x_2^\nu),$$

kde pravá strana je díky okrajové podmínce  $x_1^\nu = x_1^{\nu+1} = a$  vždy záporná a tedy  $x_i^{\nu+1} < x_{i+1}^\nu$ . Dosazením této nerovnosti do (3.58) získáme, že i  $x_{i-1}^{\nu+1} < x_i^{\nu+1}$ , tedy opět nedochází k přesmyknutí uzlů výpočetní sítě.

Při adaptaci sítě tedy opakovaně provádíme výpočet nových poloh jednotlivých uzlů dle vztahů (3.54) nebo (3.58). Tento postup se provádí tak dlouho, dokud není splněno, že  $|x_i^{\nu+1} - x_i^\nu| \leq \epsilon$  pro  $i = 0, \dots, J+1$ , kde  $\epsilon > 0$  je předepsaná tolerance, nebo, dokud není dosaženo maximálního počtu iterací (*MAXITER*). Po každé změně polohy uzlů je třeba vzít v úvahu, že je třeba vyhodnotit i monitorovací funkce. Konkrétní postupy přepočtu numerického řešení a monitorovacích funkcí budou ukázány v dalších kapitolách. Porovnání vhodnosti obou postupů pro adaptaci výpočetní sítě bude provedeno v kapitole (3.1.6).

### 3.1.5 Numerické řešení MMPDE v 2D

Pro numerické řešení rovnic (3.2), resp. (3.5) ve 2D použijeme metodu umělého času. Konkrétně budeme vycházet z následující formulace pro rovnice (3.2) [5]:

$$\begin{aligned}\frac{\partial \xi_1}{\partial \tau} &= \frac{1}{\mathcal{T} \sqrt{(g)}} \nabla^T (\mathbb{G}^{-1} \nabla \xi_1), \\ \frac{\partial \xi_2}{\partial \tau} &= \frac{1}{\mathcal{T} \sqrt{(g)}} \nabla^T (\mathbb{G}^{-1} \nabla \xi_2),\end{aligned}\tag{3.59}$$

kde  $\xi_1 = \xi_1(\mathbf{x}, \tau)$ ,  $\xi_2 = \xi_2(\mathbf{x}, \tau)$ ,  $g = \det(\mathbb{G})$  a  $\mathcal{T} > 0$  je parametr ovlivňující rychlost posunu sítě, pro malé hodnoty  $\mathcal{T}$  dochází k rychlejšímu posunu uzlů sítě. Ve shodě s literaturou budou tyto rovnice určující pohyb uzlů sítě nazývány MMPDE<sup>8</sup>. Použitím výše uvedeného převedeme řešení rovnice (3.5) na řešení následující rovnice pro  $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}, \tau)$ :

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial \tau} &= \frac{\mathbf{x}_{\xi_1}}{\mathcal{T} \mathcal{J} \sqrt{g}} \left[ \frac{\partial}{\partial \xi_1} \left( \frac{\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_2}}{\mathcal{J} g} \right) - \frac{\partial}{\partial \xi_2} \left( \frac{\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_2}}{\mathcal{J} g} \right) \right] \\ &+ \frac{\mathbf{x}_{\xi_2}}{\mathcal{T} \mathcal{J} \sqrt{g}} \left[ -\frac{\partial}{\partial \xi_1} \left( \frac{\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_1}}{\mathcal{J} g} \right) + \frac{\partial}{\partial \xi_2} \left( \frac{\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_1}}{\mathcal{J} g} \right) \right],\end{aligned}\tag{3.60}$$

kde  $\mathcal{J} = \frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_1}$  je Jacobián transformace souřadnic, u kterého předpokládáme  $\mathcal{J} \neq 0$ . Pro snazší diskretizaci převedme rovnici (3.60) na tvar:

$$\frac{\partial \mathbf{x}}{\partial \tau} = \mathbb{A} \mathbf{x}_{\xi_1 \xi_1} + \mathbb{B} \mathbf{x}_{\xi_1 \xi_2} + \mathbb{C} \mathbf{x}_{\xi_2 \xi_2} + D \mathbf{x}_{\xi_1} + E \mathbf{x}_{\xi_2}.\tag{3.61}$$

---

<sup>8</sup>Existuje více druhů pohybových rovnic pro výpočetní síť, tato formulace bývá v literatuře nazývána jako MMPDE5.

Matice  $\mathbb{A}, \mathbb{B}, \mathbb{C}$  a koeficienty  $D, E$  jsou následující:

$$\begin{aligned}
\mathbb{A} &= \frac{1}{\mathcal{T} \mathcal{J}^3 g^{3/2}} \left[ -(\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_2}) \mathbf{x}_{\xi_1} \mathbf{x}_{\xi_2}^T \mathbb{S} + \mathcal{J} \mathbf{x}_{\xi_2} \mathbf{x}_{\xi_2}^T \mathbb{G} + (\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_1}) \mathbf{x}_{\xi_2} \mathbf{x}_{\xi_2}^T \mathbb{S} \right], \\
\mathbb{B} &= \frac{1}{\mathcal{T} \mathcal{J}^3 g^{3/2}} \left[ (\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_2}) \mathbf{x}_{\xi_1} \mathbf{x}_{\xi_1}^T \mathbb{S} - \mathcal{J} \mathbf{x}_{\xi_1} \mathbf{x}_{\xi_2}^T \mathbb{G} + (\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_2}) \mathbf{x}_{\xi_1} \mathbf{x}_{\xi_2}^T \mathbb{S} \right] \\
&\quad + \frac{1}{\mathcal{T} \mathcal{J}^3 g^{3/2}} \left[ -(\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_1}) \mathbf{x}_{\xi_2} \mathbf{x}_{\xi_2}^T \mathbb{S} - \mathcal{J} \mathbf{x}_{\xi_2} \mathbf{x}_{\xi_1}^T \mathbb{G} - (\mathbf{x}_{\xi_2}^T \mathbb{G} \mathbf{x}_{\xi_1}) \mathbf{x}_{\xi_2} \mathbf{x}_{\xi_1}^T \mathbb{S} \right], \\
\mathbb{C} &= \frac{1}{\mathcal{T} \mathcal{J}^3 g^{3/2}} \left[ (\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_1}) \mathbf{x}_{\xi_2} \mathbf{x}_{\xi_1}^T \mathbb{S} + \mathcal{J} \mathbf{x}_{\xi_1} \mathbf{x}_{\xi_1}^T \mathbb{G} + (\mathbf{x}_{\xi_1}^T \mathbb{G} \mathbf{x}_{\xi_2}) \mathbf{x}_{\xi_1} \mathbf{x}_{\xi_1}^T \mathbb{S} \right], \quad (3.62) \\
D &= \frac{1}{\mathcal{T} \mathcal{J}^2 g^{1/2}} \left[ -\mathbf{x}_{\xi_2}^T \frac{\partial}{\partial \xi_1} \left( \frac{\mathbb{G}}{g} \right) \mathbf{x}_{\xi_2} + \mathbf{x}_{\xi_1}^T \frac{\partial}{\partial \xi_2} \left( \frac{\mathbb{G}}{g} \right) \mathbf{x}_{\xi_2} \right], \\
E &= \frac{1}{\mathcal{T} \mathcal{J}^2 g^{1/2}} \left[ \mathbf{x}_{\xi_2}^T \frac{\partial}{\partial \xi_1} \left( \frac{\mathbb{G}}{g} \right) \mathbf{x}_{\xi_1} - \mathbf{x}_{\xi_1}^T \frac{\partial}{\partial \xi_2} \left( \frac{\mathbb{G}}{g} \right) \mathbf{x}_{\xi_1} \right],
\end{aligned}$$

kde  $\mathbb{S}$  je matice:

$$\mathbb{S} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Buď  $\mathcal{T}_h^c$  triangulace oblasti  $\Omega_c$ , označme prostor všech po částech lineárních polynomů na triangulaci  $\mathcal{T}_h^c$  v  $\Omega_c$  v proměnných  $\xi_1$  a  $\xi_2$  jako  $\mathcal{L}_{\mathcal{T}_h^c}$  a  $\mathcal{L}_{\mathcal{T}_h^c} = [\mathcal{L}_{\mathcal{T}_h^c}]^2$ . Pro diskretizaci rovnice (3.61) je použita standardní semidiskretizace, kde pro prostorové souřadnice použijeme po částech lineární polynomy z prostoru  $\mathcal{L}_{\mathcal{T}_h^c}$  a pro časovou diskretizaci je použita zpětná Eulerova formule. Abychom se vyhnuli řešení soustavy nelineárních rovnic, tak matice  $\mathbb{A}, \mathbb{B}, \mathbb{C}$  a koeficienty  $D$  a  $E$ , viz (3.62), v časovém kroku  $\tau_\nu$  vyhodnotíme pomocí  $\mathbf{x}(\tau_\nu)$ . Výsledná soustava lineárních rovnic pro určení  $\mathbf{x}(\tau_{\nu+1}) \in \mathcal{L}_{\mathcal{T}_h^c}$  je pak následující [5]:

$$\begin{aligned}
&(\mathbf{x}_\tau(\tau_\nu), \boldsymbol{\nu}) + (\mathbf{x}_{\xi_1}(\tau_{\nu+1}), (\mathbb{A}(\tau_\nu)^T \boldsymbol{\nu})_{\xi_1}) + (\mathbf{x}_{\xi_2}(\tau_{\nu+1}), (\mathbb{C}(\tau_\nu)^T \boldsymbol{\nu})_{\xi_2}) \\
&\quad + \frac{1}{2} \left[ (\mathbf{x}_{\xi_1}(\tau_{\nu+1}), (\mathbb{B}(\tau_\nu)^T \boldsymbol{\nu})_{\xi_2}) + (\mathbf{x}_{\xi_2}(\tau_{\nu+1}), (\mathbb{B}(\tau_\nu)^T \boldsymbol{\nu})_{\xi_1}) \right] \\
&\quad - (D(\tau_\nu) \mathbf{x}_{\xi_1}(\tau_{\nu+1}) + E(\tau_\nu) \mathbf{x}_{\xi_2}(\tau_{\nu+1}), \boldsymbol{\nu}) = 0, \quad \forall \boldsymbol{\nu} \in \mathcal{L}_{\mathcal{T}_h^c}^0(\Omega_c), \quad (3.63)
\end{aligned}$$

kde  $\mathbf{x}_\tau(\tau_\nu)$  je dáno pomocí

$$\mathbf{x}_\tau(\tau_\nu) = \frac{\mathbf{x}(\tau_{\nu+1}) - \mathbf{x}(\tau_\nu)}{\tau_{\nu+1} - \tau_\nu}. \quad (3.64)$$

Symbolem  $(\cdot, \cdot)$  rozumíme skalární součin na  $L^2(\Omega_c)$  a  $\mathcal{L}_{\mathcal{T}_h^c}^0$  je podprostor  $\mathcal{L}_{\mathcal{T}_h^c}$  obsahující funkce, jež jsou na hranici  $\partial\Omega_c$  nulové. Členy  $\mathbb{A}(\tau_\nu), \mathbb{B}(\tau_\nu), \mathbb{C}(\tau_\nu), D(\tau_\nu), E(\tau_\nu)$  značí výše zmíněné matice a koeficienty vyhodnocené v časovém kroku  $\tau_\nu$ .

Poznamenejme, že obdobně jako v 1D budeme monitorovací funkci zhlazovat, tj. pro výpočet matic  $\mathbb{A}, \mathbb{B}, \mathbb{C}$  a koeficientů  $D, E$  použijeme místo monitorovací funkce  $\mathbb{G}$  námi definované zhlazení  $\tilde{\mathbb{G}}$ :

$$\tilde{\mathbb{G}}(P_i) := \frac{1}{|\mathcal{N}(P_i)| + 1} \left( \mathbb{G}(P_i) + \sum_{P_j \in \mathcal{N}(P_i)} \mathbb{G}(P_j) \right), \quad (3.65)$$

kde  $\mathcal{N}(P_i)$  značí množinu sousedních uzlů k uzlu  $P_i$  a  $|\mathcal{N}(P_i)|$  její mohutnost.

Popišme si nyní jak volit oblast  $\Omega_c$  a jak na ní získat síť konečných prvků<sup>9</sup>, která má stejnou topologii jako síť  $\mathcal{T}_h$ , jež máme k dispozici v  $\Omega$ , tj. jedná se nám o identickou síť, ale na oblasti  $\Omega_c$ . V případě použití monitorovací funkce (3.47) založené na harmonickém zobrazení a konvexnosti oblasti  $\Omega_c$  je zaručena existence a jednoznačnost řešení (viz poznámka v kapitole 3.1.3). Z tohoto důvodu, v případě konvexnosti oblasti  $\Omega$ , je nejjednodušší volit oblast  $\Omega_c = \Omega$  a použijeme přímo síť  $\mathcal{T}_h$ . V případě, že oblast  $\Omega$  není konvexní, postupujme následovně. Označme jednotlivé části hranice  $\partial\Omega$  jako  $\Gamma_i \subset \partial\Omega$ , tyto části hranice jsou buď disjunktní, nebo je jejich průnikem bod a  $\bigcup_{i=1}^n \Gamma_i = \partial\Omega$ , kde  $n$  je jejich počet. Jako  $\Omega_c$  zvolíme libovolnou konvexní oblast, která má stejný počet částí hranice  $\Gamma_i^c \subset \partial\Omega_c$ , které jsou buď disjunktní, nebo je jejich průnikem bod a  $\bigcup_{i=1}^n \Gamma_i^c = \partial\Omega_c$ . Abychom v oblasti  $\Omega_c$  získali síť, potřebujeme zobrazení  $\boldsymbol{\xi}(\mathbf{x})$  sítě z  $\Omega$  do  $\Omega_c$ . Aby tato síť byla rozmístěna rovnoměrně, řešme rovnici (3.2), kde jako monitorovací funkci  $\mathbb{G}$  použijeme jednotkovou matici  $\mathbb{I}$ , a dostaneme následující úlohu [5]:

$$\Delta \boldsymbol{\xi}(\mathbf{x}) = 0, \text{ v } \Omega, \quad (3.66)$$

s okrajovou podmínkou

$$\boldsymbol{\xi}|_{\Gamma_i} : \Gamma_i \longrightarrow \Gamma_i^c, \quad i = 1, \dots, n, \quad (3.67)$$

kde zobrazení  $\boldsymbol{\xi}|_{\Gamma_i}$  jsou vzájemně jednoznačná. Tuto úlohu řešíme numericky a získáme tak síť na oblasti  $\Omega_c$ .

Jako okrajovou podmínku pro rovnice (3.60), resp. (3.61), budeme uvažovat Dirichletovu okrajovou podmínku, protože použití Neumannovy okrajové podmínky v [5] vedlo k horšímu chování při adaptaci sítě. Triviální Dirichletova okrajová podmínka je taková, že držíme uzly sítě fixované na hranici, čili předepíšeme pro každý hraniční uzel jeho polohu na příslušné části hranice  $\Gamma_i$ .

V dalším předpokládejme, že lze příslušnou část hranice  $\Gamma_i$  jednoznačně parametrizovat pomocí funkce

$$\boldsymbol{\psi}_i(s) : [0, 1] \longrightarrow \Gamma_i. \quad (3.68)$$

Pro pohyb uzlů sítě, kde jednotlivé části hranice  $\Gamma_i$  jsou tvořeny úsečkami, použijeme námi navržený postup, který je modifikací, zjednodušením, postupu uvedeného v [5] pro obecně nelineární část hranice. Okrajovou podmínku pro příslušnou část hranice  $\Gamma_i$  určíme jako řešení 1D úlohy (3.53), kterou budeme řešit pro parametr  $s$  místo  $x$ . Místo monitorovací funkce  $G$ , použijeme její projekci  $M$  na příslušnou část hranice  $\Gamma_i$ , danou:

$$M(\boldsymbol{\psi}_i(s)) = \mathbf{t}(\boldsymbol{\psi}_i(s))^T \mathbb{G}(\boldsymbol{\psi}_i(s)) \mathbf{t}(\boldsymbol{\psi}_i(s)), \quad (3.69)$$

kde  $\mathbf{t}(\boldsymbol{\psi}_i(s))$  je jednotkový tečný vektor v daném bodě hranice. Tento postup vede k tomu, že v  $M$  se projeví jen některé prvky monitorovací funkce  $\mathbb{G}$ , v závislosti na tečném vektoru  $\mathbf{t}$  hranice  $\Gamma_i$ . Takto získané nové polohy hraničních uzlů předepíšeme jako okrajové podmínky pro numerické řešení úlohy (3.60), resp. (3.61).

Poznamenejme, že při každé změně sítě v  $\Omega$ , je nutné aktualizovat numerické řešení  $\mathbf{w}$ , respektive funkci  $u$ , které na této síti máme. Při této aktualizaci dojde nutně ke změně hodnoty monitorovací funkce  $\mathbb{G}$  a matic a koeficientů (3.62).

<sup>9</sup>Pro numerické řešení rovnice (3.63) na oblasti  $\Omega_c$  používáme metodu konečných prvků, zatímco na oblasti  $\Omega$  máme síť konečných objemů  $\mathcal{T}_h$ , kterou chceme adaptovat.

## Postup řešení v 2D

V dalším předpokládejme, že na každém z konečných objemů  $D_i$  máme k dispozici po částech konstantní aproximaci veličiny  $u$ , a označme ji  $u_i$ . Dále předpokládejme, že na oblasti  $\Omega_c$  máme k dispozici síť se stejnou topologií jako v  $\Omega$ . Vlastní algoritmus pro řešení rovnic vypadá následovně<sup>10</sup>:

### Algoritmus 3.1. Algoritmus adaptace sítě

1. Z po částech konstantních hodnot  $u_i$  zkonstruujeme po částech lineární funkci  $u_h^*$  tak, že určíme její hodnoty v uzlech sítě  $P_j$  pomocí váženého průměru:

$$u_h^*(P_j) = \frac{\sum_{D_i \in \mathcal{K}_{P_j}} |D_i| u_i}{\sum_{D_i \in \mathcal{K}_{P_j}} |D_i|}, \quad (3.70)$$

kde  $\mathcal{K}_{P_j}$  označuje množinu objemů, které mají společný uzel  $P_j$ .

2. Na oblasti  $\Omega_c$  aproximujeme  $\nabla u$  a spočteme hodnotu monitorovací funkce  $\mathbb{G}$  v uzlech sítě.
3. V případě, že umožníme pohyb uzlů po hranici řešíme 1D úlohu (3.53) pro parametr  $s$  s využitím projekce (3.69) a získáme předpis okrajových podmínek na přílušné části hranice  $\Gamma_i^c$ . V opačném případě předepíšeme pro každý hraniční uzel z  $\Gamma_i^c$  jeho polohu na přílušné části hranice  $\Gamma_i$ .
4. Na základě řešení z bodu 3. provedeme posun uzlů sítě nacházejících se na částech hranice  $\Gamma_i$ . Provedeme přepočítání veličiny  $u$  na novou adaptovanou síť a určíme hodnotu monitorovací funkce  $\mathbb{G}$  v uzlech nové sítě.
5. Počáteční podmínku určíme tak, že každému z uzlů výpočetní sítě na  $\Omega_c$  přiřadíme jeho počáteční polohu na síti v  $\Omega$ .
6. Spočteme matice  $\mathbb{A}, \mathbb{B}, \mathbb{C}$  a koeficienty  $D, E$  a řešíme úlohu danou (3.63) za použití okrajových podmínek určených v kroku 3., tím získáme  $\mathbf{x}(\tau_{\nu+1})$ . Zvýšíme časový krok  $\nu$  a tento bod opakujeme pro předepsaný maximální počet iterací (*TSTEP*), nebo do té doby než dosáhneme předepsané odchylky  $\epsilon$  od stacionárního stavu, čili je splněno<sup>11</sup>:

$$REZ = \sqrt{\int_{\Omega_c} (\mathbf{x}(\boldsymbol{\xi}, \tau_{\nu+1}) - \mathbf{x}(\boldsymbol{\xi}, \tau_\nu))^2 d\xi_1 d\xi_2} \leq \epsilon \quad (3.71)$$

7. Na základě spočtených hodnot  $\mathbf{x}(\boldsymbol{\xi}, \tau_\nu)$  provedeme adaptaci, čili zobrazení sítě z  $\Omega_c$  do  $\Omega$  a přepočteme veličinu  $u$  na novou adaptovanou síť.
8. V případě, že jsme nedosáhli předepsaného počtu iterací (*MAXITER*), znovu spočteme monitorovací funkci  $\mathbb{G}$  a pokračujeme bodem 6.

Z výše uvedeného algoritmu je vidět, že adaptace sítě se provádí každých *TSTEP* kroků, důvodem je to, že vlastní adaptace sítě (zobrazení z  $\Omega_c$  do  $\Omega$ ) a následný přepočítání hodnot  $u$  a monitorovací funkce  $\mathbb{G}$  je časově dosti náročný.

<sup>10</sup>Autoři v článku [5], ze kterého jsme vycházeli, detailně nespecifikují použitý algoritmus řešení.

<sup>11</sup>Ukazatel *REZ* má tedy tento význam  $REZ = \|\mathbf{x}^{\nu+1} - \mathbf{x}^\nu\|_{L^2(\Omega_c)}$ .

## Ukázková úloha

Protože výše uvedený algoritmus je poměrně komplikovaný, ilustrujme si jej na ukázkové úloze. Volme oblast  $\Omega$  jako lichoběžník daný body  $A, B, C, D$ , kde

$$A = [-5, -5], B = [5, -5], C = [2.5, 2.5], D = [-2.5, 2.5].$$

Přesto, že je oblast  $\Omega$  konvexní, volme jako oblast  $\Omega_c$  jednotkový čtverec daný body  $A_c, B_c, C_c, D_c$ , kde

$$A_c = [0, 0], B_c = [1, 0], C_c = [1, 1], D_c = [0, 1].$$

Jednotlivé části hranice  $\Gamma_i$ , resp.  $\Gamma_i^c$  jsou následující úsečky:

$$\Gamma_1 = AB, \Gamma_1^c = A_c B_c$$

$$\Gamma_2 = BC, \Gamma_2^c = B_c C_c$$

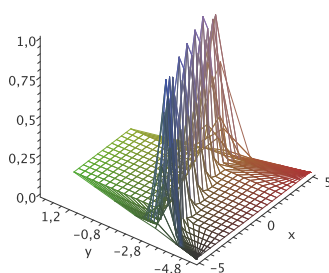
$$\Gamma_3 = CD, \Gamma_3^c = C_c D_c$$

$$\Gamma_4 = DA, \Gamma_4^c = D_c A_c.$$

Síť  $\mathcal{T}_h$  na oblasti  $\Omega$  má 3188 objemů a 1675 uzlů. Použijme pro adaptaci sítě monitorovací funkci Winslowova typu (3.43) s  $\alpha = 1$  a v dalším předpokládejme, že funkce  $u$ , jejíž gradient potřebujeme pro konstrukci  $\mathbb{G}$ , je dána předpisem:

$$u(x, y) = e^{-(x-4(y+2))^2}.$$

Z průběhu funkce  $u$  zaneseném na obrázku 3.9 je patrné, že  $u(x, y) = 1$  na přímce  $y = x/4 - 2$  a s rostoucí vzdáleností od této přímky velmi prudce klesá. Parametr rychlosti posunu sítě  $\mathcal{T}$ , viz (3.59), a velikost časového kroku  $\Delta\tau = \tau_{\nu+1} - \tau_\nu$ , viz (3.64), volíme  $\mathcal{T} = 0.01$  a  $\Delta\tau = 0.001$ . Okrajové podmínky stanovíme takové, že na  $\Gamma_1$  a  $\Gamma_3$  jsou polohy hraničních uzlů fixované a na  $\Gamma_2$  a  $\Gamma_4$  umožníme pohyb uzlů po příslušné části hranice.



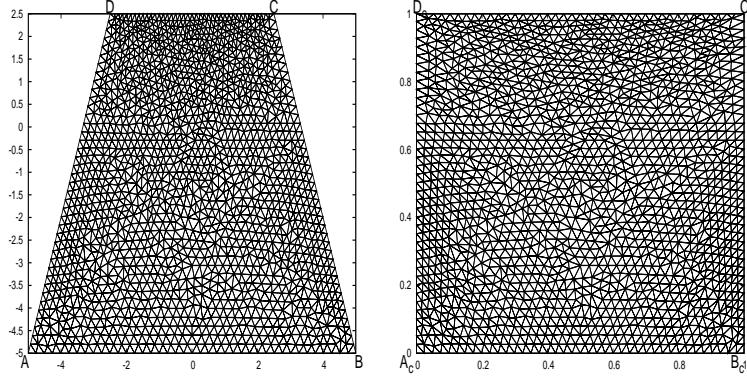
Obrázek 3.9: Průběh funkce  $u$  na oblasti  $\Omega$ .

Poznamenejme, že všechny výpočty pomocí metody konečných prvků realizujeme pomocí nástroje FreeFem++<sup>12</sup>, který je určen pro výpočty pomocí metody konečných prvků ve 2D.

<sup>12</sup>Pro vlastní numerické řešení používáme nástroj FreeFem++ verze 3.40, [24], dostupné na <http://www.freefem.org>



Nejprve se zaměříme na získání výpočetní sítě se stejnou topologií na oblasti  $\Omega_c$ . Předepíšeme okrajové podmínky tak, aby platila podmínka (3.67), v našem případě volíme  $\xi|_{\Gamma_i}$  ve tvaru  $\xi|_{\Gamma_i}(\mathbf{x}) = \mathbb{P}_i \mathbf{x}$ ,  $\mathbf{x} \in \Gamma_i$ , kde  $\mathbb{P}_i \in \mathbb{R}^{2 \times 2}$  je matice. Poté řešíme úlohu (3.66) pro nalezení zobrazení  $\xi(\mathbf{x}) : \Omega \rightarrow \Omega_c$ . Původní síť v  $\Omega$  a její obraz v  $\Omega_c$  jsou zaneseny na obrázku 3.10.



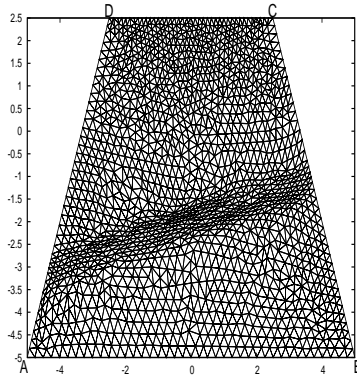
Obrázek 3.10: Původní síť v  $\Omega$  (vlevo) a její obraz v  $\Omega_c$  (vpravo).

Vlastní adaptaci sítě začneme přímo bodem 2., neboť funkci  $u$  máme předepsanou a FreeFem++ ji přímo převede na konečně prvkovou reprezentaci, která je po částech lineární.

Po spočtení monitorovací funkce  $\mathbb{G}$  (bod 2.) určíme její projekci  $M$  (3.69) na  $\Gamma_2$  a  $\Gamma_4$  a na těchto částech hranice řešíme úlohu (3.53) pro parametr  $s$  za použití schématu (3.58), (bod 3.). Použité parametrizace (3.68) hranic  $\Gamma_2$ , resp.  $\Gamma_4$  jsou  $\psi_2(s) = B + (C - B)s$ ,  $s \in [0, 1]$ , resp.  $\psi_4(s) = D + (A - D)s$ ,  $s \in [0, 1]$ . Provedeme posuv hraničních uzlů na částech hranice  $\Gamma_2$  a  $\Gamma_4$ , na částech hranice  $\Gamma_1$  a  $\Gamma_3$  jsou polohy hraničních uzlů fixované, a přepočteme hodnoty  $u$  a monitorovací funkce  $\mathbb{G}$  (bod 4.).

Každému uzlu  $P_i$  na síti v  $\Omega_c$  přiřadíme jeho souřadnice na síti  $\mathcal{T}_h$  v  $\Omega$ , čili předepíšeme počáteční podmínku  $\mathbf{x}(\tau_0)$  (bod 5.).

Řešení z bodu 3. použijeme jako okrajovou podmínku v kroku 6.<sup>13</sup>, kde jsme volili  $TSTEP = 1$ . Pokračujeme body 7. a 8. a toto celé je *MAXITER*-krát zopakováno (v našem případě *MAXITER* = 40). Výsledná adaptovaná síť je zanesena na následujícím obrázku 3.11.



Obrázek 3.11: Výsledná síť na oblasti  $\Omega$ .

<sup>13</sup>Opět používáme nástroj FreeFem++.

### 3.1.6 Numerické experimenty v 1D

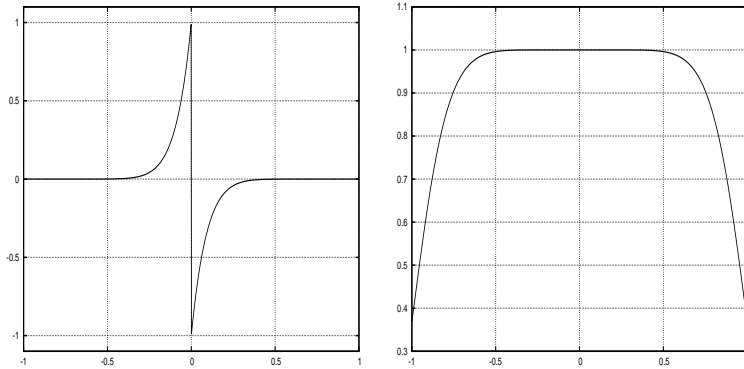
V této kapitole se zaměříme na porovnání 1D algoritmů s umělým časem (3.54) a s G-S iterací (3.58) pro adaptaci výpočetní sítě vycházejících z variačních metod, jedná se o námi provedené numerické experimenty. Pro studium samotného adaptačního procesu jsme se v této fázi zaměřili na úlohu, kde veličinu  $u(x)$  máme předepsanou. Budeme zde též zjišťovat vliv parametru  $\alpha$  při výpočtu monitorovací funkce (3.44) na proces adaptace. Důvodem k tomu je, že samotná adaptace sítě je součástí komplikovanějších algoritmů a proto, dříve než se jim budeme v kapitole 4 věnovat, potřebujeme rozumět chování procesu adaptace<sup>14</sup>.

Jako funkce  $u(x)$  byly vybrány následující dvě funkce:

$$u_A(x) = \begin{cases} x < 0 & (x + 1)^{11}, \\ x \geq 0 & (x - 1)^{11}, \end{cases} \quad (3.72)$$

$$u_B(x) = e^{-x^8}. \quad (3.73)$$

Vidíme, že v prvním případě (3.72) se jedná o úlohu s nespojitostí a ve druhém



Obrázek 3.12: Průběhy funkcí  $u_A(x)$  a  $u_B(x)$  na  $[-1, 1]$ .

případě (3.73) se jedná o úlohu, kde je oblast s téměř konstantní hodnotou. Jako fyzikální oblast  $\Omega$  volíme interval  $[-1, 1]$ .

V dalším předpokládejme, že máme k dispozici po částech konstantní aproximace  $u(x)$ :

$$u_{i+\frac{1}{2}}^\nu = \frac{1}{x_{i+1}^\nu - x_i^\nu} \int_{x_i^\nu}^{x_{i+1}^\nu} u(x) dx, \quad (3.74)$$

kde  $x_i^\nu$ ,  $i = 1, \dots, J$  jsou uzlové body sítě v čase  $\tau_\nu$ . Tyto hodnoty používáme pro výpočet monitorovacích funkcí  $G(u_{i+\frac{1}{2}}^\nu)$ , kde uvažujeme monitorovací funkce Winslowova typu (3.44) v 1D a používáme zhlazení monitorovací funkce (3.55). Po posunu uzlů sítě je třeba nejprve přepočítat aproximace  $u_{i+\frac{1}{2}}^{\nu+1}$  v novém časovém kroku  $\nu + 1$ , k čemuž používáme opět vztahu (3.74), kde je daný integrál určen numericky<sup>15</sup>.

<sup>14</sup>V kapitole 4.3 tuto funkci  $u$  konstruujeme z numerického řešení rovnice (1.17) v 1D, zde si ji přímo předepíšeme. Důvodem je, že chceme studovat pouze proces adaptace bez zkraslení dalšími vlivy.

<sup>15</sup>Byla zde použita rutina implementačního jazyka Octave *quad*, což vede k volání metody založené na Gaussově kvadratuře.

Pro zhodnocení toho, jak je daná síť adaptována, předpokládejme, že na této síti chceme funkci  $u(x)$  aproximovat pomocí po částech konstantních hodnot  $u_{i+\frac{1}{2}}^\nu$ , kvalitu adaptace pak budeme sledovat pomocí následujícího ukazatele chyby<sup>16</sup>:

$$L2E = \frac{1}{J} \sum_{i=1}^J \int_{x_i}^{x_{i+1}} \left( u(x) - u_{i+\frac{1}{2}}^\nu \right)^2 dx, \quad (3.75)$$

kde integrál na pravé straně (3.75) je určen pro funkci  $u_A(x)$  analyticky a pro funkci  $u_B(x)$  pomocí numerické kvadratury<sup>17</sup>.

Volba časového kroku pro schéma (3.54) je prováděna v každé iteraci. Z podmínky stability (3.56) dostáváme pro volbu časového kroku:

$$\Delta\tau = \frac{CFL h^2}{2 \max_{i=1, \dots, J-1} (G(u_{i+\frac{1}{2}}^\nu))}, \quad (3.76)$$

kde konstanta  $CFL$  byla zvolena  $CFL = 0.9$ .

Pomocí posunu uzlů  $x_i$  mezi  $\nu$ -tou a  $\nu + 1$  iterací je sledováno přiblížení se ke stacionárnímu stavu rovnice (3.53), tedy:

$$REZ = \max_{i=1, \dots, J+1} (|x_i^{\nu+1} - x_i^\nu|). \quad (3.77)$$

Pro zastavení byla použita podmínka  $REZ < \epsilon$ , kde  $\epsilon = 10^{-6}$ . Počet objemů byl zvolen  $J = 21$ , což je pro ověření chování metody dostačující. Na počáteční síti měly všechny objemy stejnou velikost. V obou experimentech byl nastaven maximální počet iterací  $MAXITER = 30$ .

Celý adaptivní proces pro tento test byl naprogramován v jazyce pro numerické výpočty Octave<sup>18</sup>.

## Úloha typu A

Pro různé hodnoty  $\alpha$  byly sledovány výše uvedené parametry, v tabulkách je zanesena závěrečná hodnota, doba potřebná pro adaptaci<sup>19</sup> ( $T$ ) a počet skutečně provedených iterací ( $ITER$ ), než došlo k zastavení algoritmu.

$L2E$	$REZ$	$T$ [s]	$ITER$	$\alpha$
0.010686	0.000638	3.36	30	0.1
0.007495	0.002444	3.31	30	1
0.007171	0.002489	3.41	30	2
0.006828	0.002439	3.43	30	10

Tabulka 3.1: Úloha A. Použití umělého času

Pro vizuální zhodnocení vlivu adaptace sítě nejprve zkonstruujeme po částech lineární rekonstrukce funkce  $u(x)$  z hodnot  $u_{i+\frac{1}{2}}$  pomocí vztahu:

$$u^*(x) = u_{i+\frac{3}{2}} \frac{x - x_{i+\frac{1}{2}}}{x_{i+\frac{3}{2}} - x_{i+\frac{1}{2}}} + u_{i+\frac{1}{2}} \frac{x_{i+\frac{3}{2}} - x}{x_{i+\frac{3}{2}} - x_{i+\frac{1}{2}}}, \quad x \in \langle x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}} \rangle, \quad (3.78)$$

<sup>16</sup>Jedná se v podstatě o chybu měřenou v L2 normě vztaženou na jeden objem.

<sup>17</sup>Opět byla použita rutina jazyka Octave *quad*.

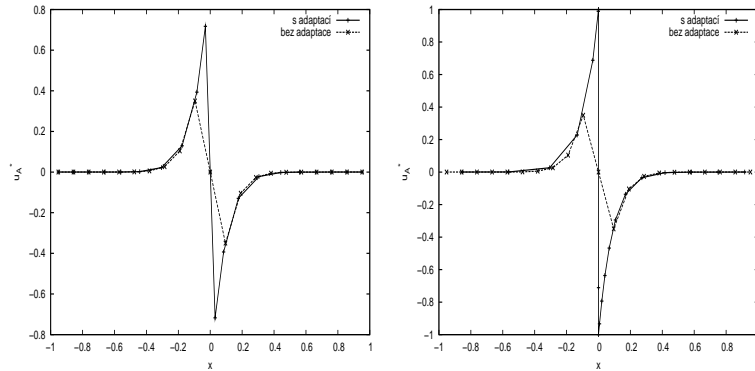
<sup>18</sup>Dostupný na : <http://www.gnu.org/software/octave/>

<sup>19</sup>Byly použity standardní rutiny v Octave. Jedná se o *tic*, *toc*

$L2E$	$REZ$	$T$ [s]	$ITER$	$\alpha$
0.009724	0.001502	3.98	30	0.1
0.002820	0.028697	3.84	30	1
0.003605	0.043161	3.78	30	2
0.001865	0.016885	3.77	30	10

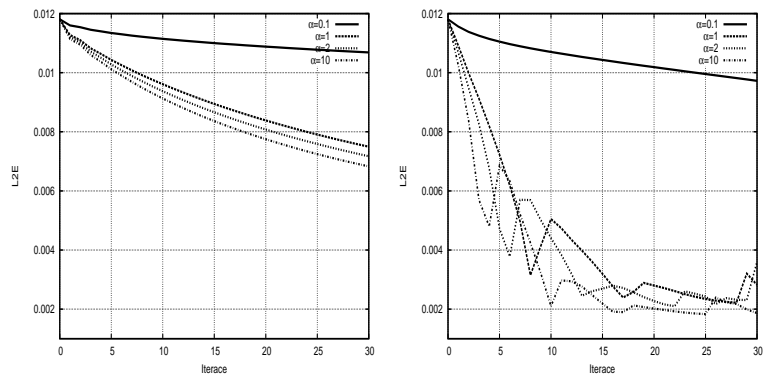
Tabulka 3.2: Úloha A. Použití G-S iterací

kde  $x_{i+\frac{1}{2}} = (x_{i+1} + x_i)/2$ , resp.  $x_{i+\frac{3}{2}} = (x_{i+2} + x_{i+1})/2$ . Porovnání těchto rekonstrukcí pro funkci  $u_A(x)$ , na adaptované a původní síti pro obě metody s hodnotou  $\alpha = 1$ , je na obrázku 3.13. Zde je vidět, že při použití adaptace sítě výsledná rekonstrukce lépe zachycuje nespojitost, než bez použití adaptace sítě, kde jsou uzlové body rozmístěny rovnoměrně. Dále je vidět, že ostřeji je nerovnost zachycena na obrázku vpravo, kde je použito G-S iterací (3.58), toto je způsobeno tím, že do blízkosti nespojitosti se za použití této metody dostal větší počet bodů, viz trajektorie jednotlivých síťových bodů na obrázku 3.16.

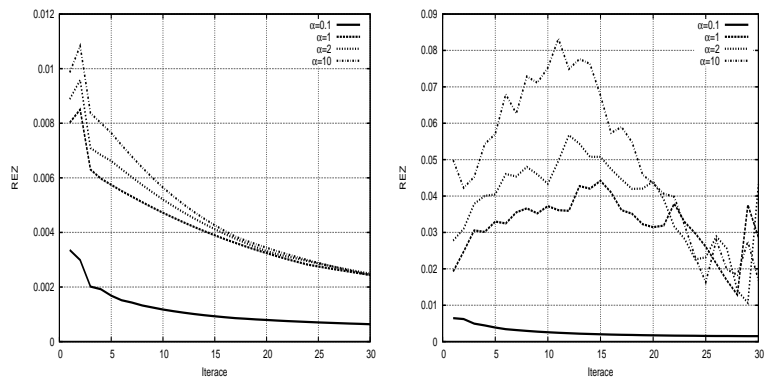


Obrázek 3.13: Úloha A. Rekonstrukce  $u_A^*(x)$  pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo), pro obě metody je použito  $\alpha = 1$ , v grafu jsou vyznačeny hodnoty  $x_{i+\frac{1}{2}}$ .

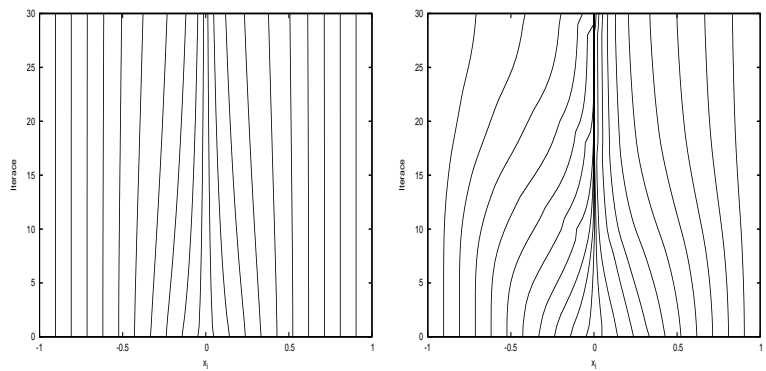
Průběhy jednotlivých sledovaných veličin udávají grafy 3.14 a 3.15. Z průběhů sledovaných veličin je vidět, že při použití G-S iterací dochází při relativně malém počtu iterací k rychlému poklesu sledovaných veličin, avšak při větším počtu iterací dochází k rozkmitání uzlů sítě v závislosti na  $\alpha$ . Rozkmitání uzlů sítě je též patrné z obrázku 3.16(vpravo), kde je vidět, že pro několik posledních iterací dochází k odsunu uzlů od místa nespojitosti. Důvodem tohoto rozkmitání je vyšší citlivost schématu (3.58) na skokové změny monitorovací funkce  $G$ . Použitím vícenásobného zhlazení dle vztahu (3.55) lze rozkmitání zabránit. Z výše uvedeného je vidět, že zvýšení hodnoty  $\alpha$  vede k větší citlivosti metod. Dále je z 3.14 vidět, že adaptace sítě vedla ke zmenšení chyby, které se dopouštíme při aproximaci funkce  $u_A(x)$  po částech konstantní hodnotou, ve smyslu ukazatele  $L2E$  (3.75).



Obrázek 3.14: Úloha A. Průběh L2E pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo).



Obrázek 3.15: Úloha A. Průběh REZ (3.77) pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo).



Obrázek 3.16: Úloha A. Trajektorie jednotlivých uzlů sítě, pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo)

## Úloha typu B

Protože nyní uvažovaná funkce  $u_B(x)$  obsahuje oblast s velmi malou hodnotou derivace, byly hodnoty  $\alpha$  zvoleny v jiném rozsahu než v předešlém případě. Pro výpočet veličiny  $L2E(3.75)$  byla použita numerická kvadratura<sup>20</sup>, hodnota odhadu chyby kvadratury na každém z objemů byla řádově  $10^{-15}$ .

$L2E$	$REZ$	$T$ [s]	$ITER$	$\alpha$
0.001910	0.000343	7.61	30	0.1
0.001502	0.001981	7.66	30	1
0.001331	0.003887	7.51	30	10
0.001278	0.005291	7.51	30	100

Tabulka 3.3: Úloha B. Použití umělého času

$L2E$	$REZ$	$T$ [s]	$ITER$	$\alpha$
0.001867	0.000206	7.53	30	0.1
0.001261	0.001710	7.45	30	1
0.000952	0.004944	7.49	30	2
0.000869	0.004291	7.53	30	10

Tabulka 3.4: Úloha B. Použití G-S iterací

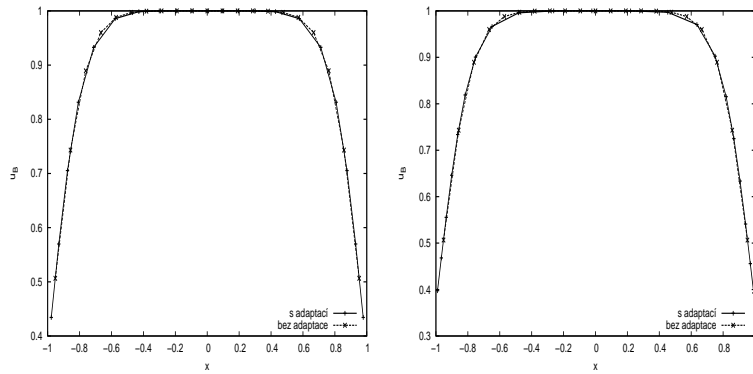
Rozdíl hodnoty  $T$  mezi současnou (viz tabulky 3.3 a 3.4) a předchozí úlohou (tabulky 3.1 a 3.2) je způsoben tím, že v tomto případě (B) je hodnota  $L2E$  určována numericky, což vede ke zpomalení výpočtu. Porovnání lineární rekonstrukce (3.78) pro funkci  $u_B(x)$ , na adaptované a původní síti pro obě metody s hodnotou  $\alpha = 10$ , je na obrázku 3.17. Zde je vidět, že jsou téměř identické, což je způsobeno tím, že aproximace funkce  $u_B(x)$  není tolik citlivá na volbu uzlů sítě, tj. neobsahuje skok.

Na obrázcích 3.18 a 3.19 je zanesen průběh námi sledovaných parametrů. Je vidět, že i zde došlo pro velké hodnoty parametru  $\alpha$  k částečnému rozkmitání ukazatele  $REZ$  (3.77), které je ovšem mnohem menší než v úloze předcházející. Dále je z 3.18 vidět, že adaptace sítě opět vedla ke zmenšení chyby, které se dopouštíme při aproximaci funkce  $u_B(x)$  po částech konstantní hodnotou, ve smyslu ukazatele  $L2E$  (3.75).

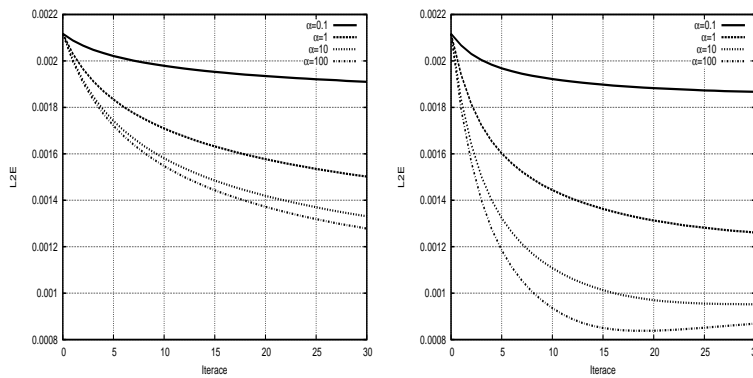
## Zhodnocení provedených experimentů v 1D

V obou provedených experimentech se ukazuje, že se zvětšováním hodnoty parametru  $\alpha$ , což je uživatelem zadávaný parametr, dochází k většímu pohybu uzlů sítě. Ze získaných dat vyplývá, že při stejné hodnotě  $\alpha$  je posun uzlů sítě výraznější při použití G-S iterací. Obdobně pokles veličiny  $L2E$  je, při stejném  $\alpha$ , pro několik prvních iterací mnohem rychlejší při použití G-S iterací. Nevýhodou této metody (G-S iterací) je, že vykazuje vyšší citlivost na změny monitorovací funkce  $G$  a tak může docházet k rozkmitávání uzlů sítě po přepočtu  $G$ , zejména pro vyšší hodnoty parametru  $\alpha$ . Z praktického hlediska, kdy budeme řešit problémy, kde je

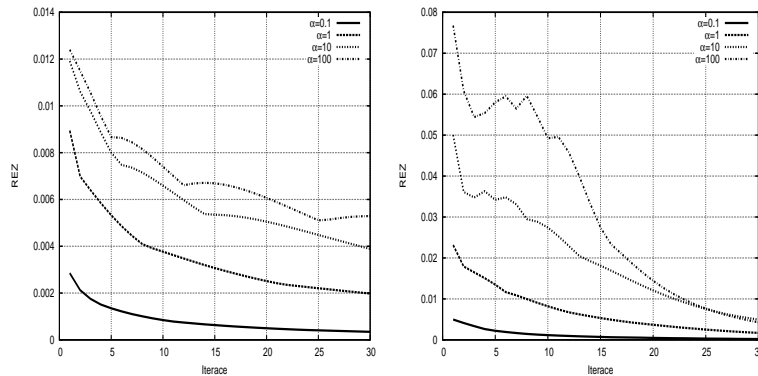
<sup>20</sup>Byla zde použita standardní rutina Octave *quad*, která používá Gaussovu kvadraturu.



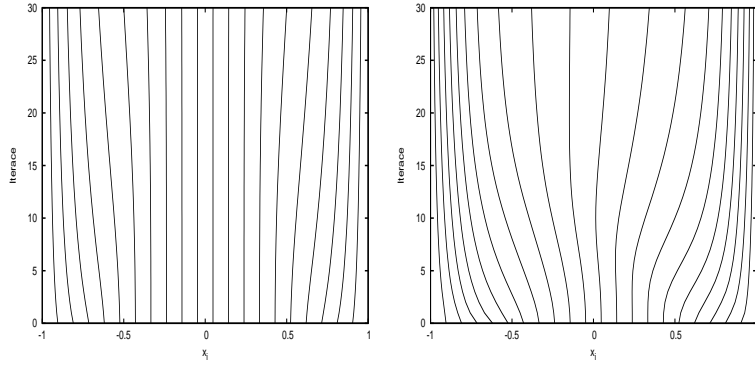
Obrázek 3.17: Úloha B. Rekonstrukce  $u_B^*(x)$  pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo), pro obě metody je použito  $\alpha = 10$ , v grafu jsou vyznačeny hodnoty  $x_{i+\frac{1}{2}}$ .



Obrázek 3.18: Úloha B. Průběh L2E pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo).



Obrázek 3.19: Úloha B. Průběh REZ (3.77) pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo).



Obrázek 3.20: Úloha B. Trajektorie jednotlivých uzlů sítě, pro metodu umělého času (vlevo) a při použití G-S iterací (vpravo).

pohybující se diskontinuita a tedy budeme často měnit polohu uzlů sítě, se jeví výhodnější použití menšího počtu G-S iterací. V případě, že dojde k rozkmitávání uzlů sítě, což lze zjistit dle parametru  $REZ$  (3.77), je možné adaptivní proces ukončit nebo zmenšit hodnotu  $\alpha$ . Popřípadě použít vícenásobné zhlazení (3.55).

### 3.1.7 Numerické experimenty ve 2D

V této části naší práce se budeme věnovat testu 2D algoritmu pro adaptaci nestrukturované výpočetní sítě pomocí variačních metod. Zaměříme se na porovnání vlivu použitých monitorovacích funkcí, dále na vliv parametrů algoritmu 3.1,  $MAXITER$  a  $TSTEP$ , tak i na vliv okrajových podmínek, na adaptaci sítě. Důvodem je to, že než danou metodu použijeme jako součást komplexnějšího algoritmu, je nutné zjistit vliv jednotlivých parametrů na adaptační proces. Poznamenejme, že zde uváděné numerické experimenty jsou původní.

Opět jako v kapitole 3.1.6 se zaměříme na úlohu, kdy veličinu  $u(\mathbf{x})$ <sup>21</sup> máme na oblasti  $\Omega$  předepsanou. Jako testovací úlohy zvolme následující dvě funkce:

$$u_A(x, y) = e^{-x^2}, \quad (3.79)$$

$$u_B(x, y) = e^{\cos(xy/2)}, \quad (3.80)$$

jejichž průběhy jsou ukázány na obrázcích 3.21 a 3.22.

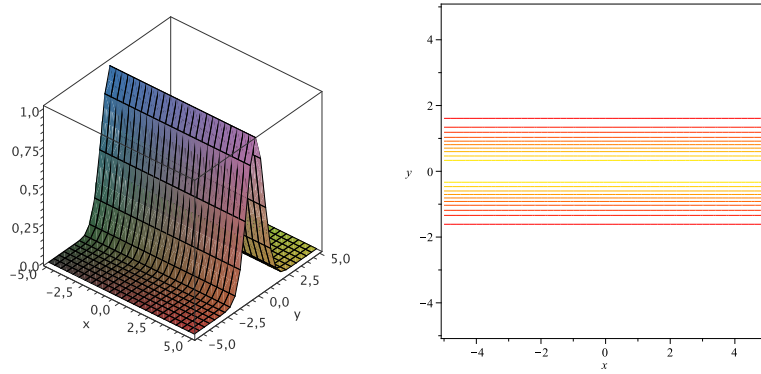
Otestujeme algoritmus pro 2D uvedený v 3.1.5 a to tak, že  $\Omega = \Omega_c$ . Jako oblasti  $\Omega$  a  $\Omega_c$  uvažujeme oblast  $[-5, 5] \times [-5, 5]$ , triangulace  $\mathcal{T}_h$ , která má v obou oblastech stejnou topologii, obsahuje 9908 objemů a 5084 uzlů a byla zkonstruována pomocí generátoru sítě v FreeFem++.

Vlastní diskretizace daná (3.63) byla implementována v jazyce FreeFem++ verze 3.40<sup>22</sup>, jako řešič systému (3.63) byl použit řešič FreeFem++, což je metoda GMRES (Generalized Minimal Residual Method) [24]. Z důvodu použitého implemetačního jazyka v této části též neuvažujeme přepočet konečně objemového řešení na po částech lineární řešení bod 1. algoritmu 3.1, vztah (3.70), do

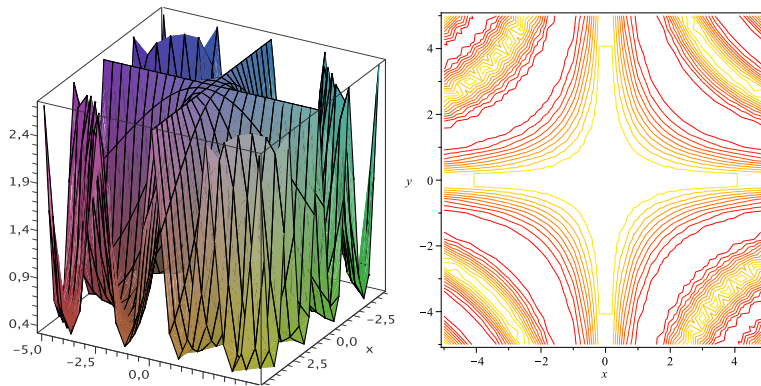
<sup>21</sup>Protože chceme studovat vlastní adaptační proces bez zkreslení dalšími vlivy, tak zde tuto funkci přímo předepisujeme. V další části práce, kapitola 4.4, bude zkonstruována z numerického řešení Eulerových rovnic (1.17).

<sup>22</sup>Dostupné na <http://www.freefem.org/>





Obrázek 3.21: Průběh funkce  $u_A(x, y)$  a její izokřivky na  $[-5, 5] \times [-5, 5]$ .



Obrázek 3.22: Průběh funkce  $u_B(x, y)$  a její izokřivky na  $[-5, 5] \times [-5, 5]$ .

programu nám vstupuje přímo zadaná funkce  $u_A$ , respektive  $u_B$  a je přepočtena přímo FreeFem++ na P1 elementy. Dále poznamenejme, že určení odchylky od stacionárního stavu (3.71), tedy hodnota  $REZ$ , je též provedeno pomocí FreeFem++, kde je daný integrál spočten numericky pomocí Gaussovy formule 6-tého řádu [24]. Hodnotu  $\epsilon$ , tedy minimální hodnotu  $REZ$  (3.71), volme  $\epsilon = 10^{-6}$ .

Nejprve se zaměříme na úlohy, kdy budeme držet uzly na hranici  $\partial\Omega$  fixované. Tedy vynecháváme z testů bod (3) algoritmu 3.1, vliv pohybu uzlů na hranici  $\Omega$  bude dále diskutován.

### Úloha typu A s fixovanými hraničními uzly

Nejprve použijme pro naši úlohu monitorovací funkci Winslowova typu, danou (3.43) a (3.44), a otestujme vliv parametrů  $TSTEP$  a  $MAXITER$  na adaptační proces. Hodnota  $\alpha$  pro monitorovací funkci Winslowova typu je volena  $\alpha = 1$  a velikost časového kroku umělého času  $\Delta\tau = \tau_{\nu+1} - \tau_\nu$ , viz (3.64), volme  $\Delta\tau = 1$ , parametr ovlivňující rychlost adaptace sítě  $\mathcal{T}$ , viz (3.59), je volen  $\mathcal{T} = 1$ . Získané hodnoty jsou zaneseny v tabulce 3.5 a průběh ukazatele dosažení stacionárního stavu  $REZ$  (3.71) na obrázku 3.23 vlevo.

Jako další otestujme pro naši úlohu monitorovací funkci založenou na harmonickém zobrazení danou vztahy (3.47) a (3.49). V dalším textu monitorovací

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.001791	151.02
2	10	0.003013	151.68
4	5	0.005256	153.70
5	4	0.003857	153.50
10	2	0.021128	156.02
20	1	0.013293	162.02

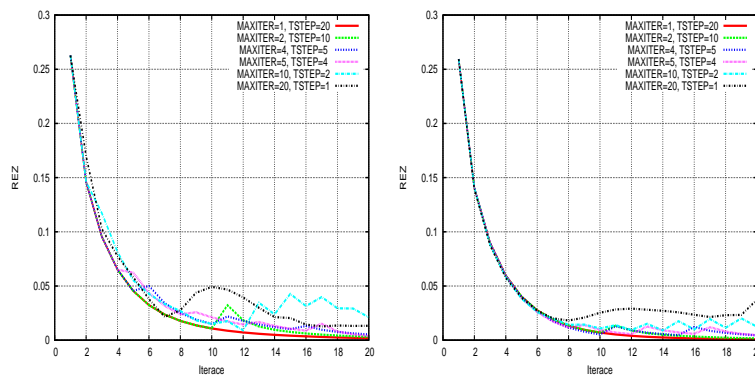
Tabulka 3.5: Úloha typu A. Sledované veličiny při  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ , použita Winslowova metoda.

funkci tohoto typu nazvěme ALLMF<sup>23</sup>. Opět, jako v předešlém případě, je voleno  $\Delta\tau = 1$  a parametr ovlivňující rychlost adaptace sítě  $\mathcal{T} = 1$ . Získané hodnoty jsou uvedeny v tabulce 3.6 a průběh sledovaných veličin během adaptačního procesu je zanesen v grafech na obrázcích 3.23 vpravo.

Z grafů 3.23 pro průběh hodnoty  $REZ$  je vidět, že každá adaptace sítě, vede k mírnému zhoršení ukazatele dosažení stacionárního stavu  $REZ$  (3.71), který však ve většině případů, v dalších krocích opět klesá.

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.000763	152.46
2	10	0.001597	152.23
4	5	0.004461	153.34
5	4	0.004667	154.10
10	2	0.012392	156.97
20	1	0.037438	162.45

Tabulka 3.6: Úloha typu A. Sledované veličiny při  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ , použita monitorovací funkce ALLMF.

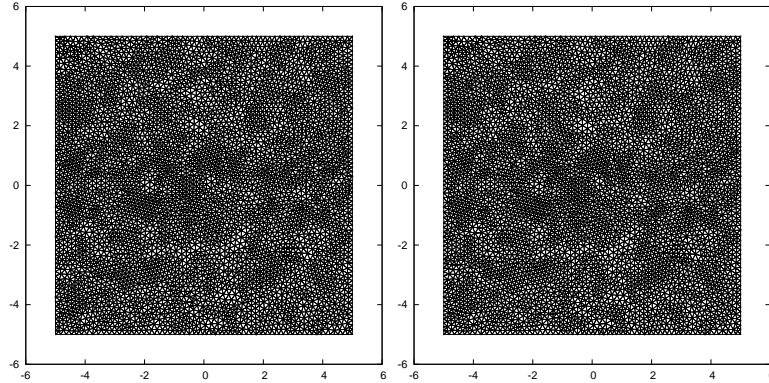


Obrázek 3.23: Úloha typu A. Průběh ukazatele  $REZ$  při použití Winslowovy metody (vlevo) a monitorovací funkce ALLMF (vpravo),  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ .

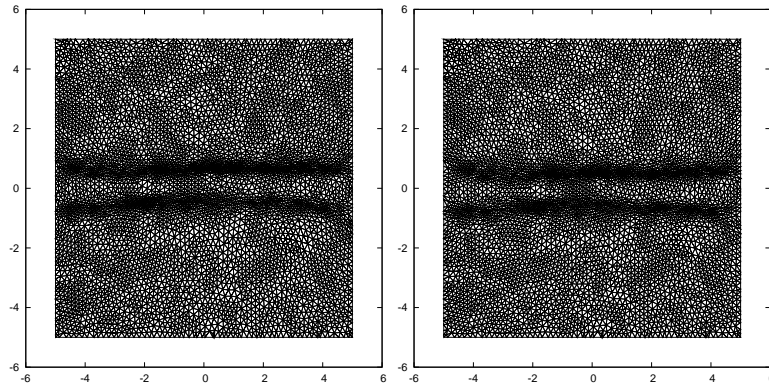
Výsledné sítě pro obě volby monitorovací funkce jsou na obrázcích 3.24, 3.25 a 3.26. Zde je vidět, že s rostoucím počtem kroků adaptace sítě, což určuje parametr  $MAXITER$ , viz algoritmus 3.1, je síť opticky výrazněji adaptována. Pro

<sup>23</sup>ARC-LENGTH LIKE MONITOR FUNCTION [6]

oba použité typy monitorovacích funkcí vycházejí výsledné sítě zhruba stejné. Z obrázku 3.26 je patrné, že na levém a pravém okraji oblasti  $\Omega$ , je výsledná síť ovlivněna tím, že jsou polohy uzlů na této hranici fixovány.



Obrázek 3.24: Úloha typu A. Adaptovaná síť při použití Winslowovy metody (vlevo) a monitorovací funkce ALLMF (vpravo),  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ ,  $MAXITER = 1$ ,  $TSTEP = 20$ .

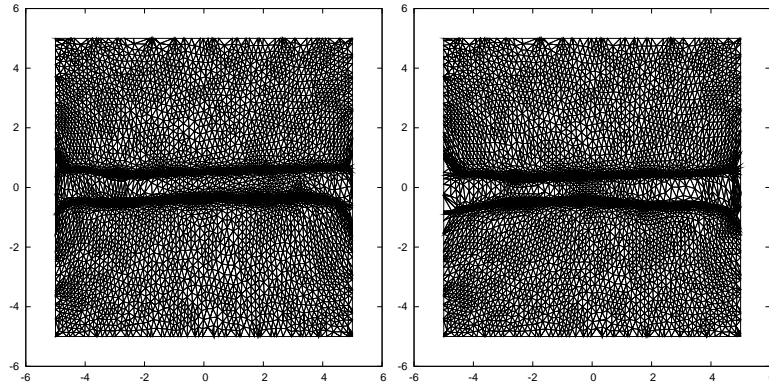


Obrázek 3.25: Úloha typu A. Adaptovaná síť při použití Winslowovy metody (vlevo) a monitorovací funkce ALLMF (vpravo),  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ ,  $MAXITER = 5$ ,  $TSTEP = 4$ .

### Úloha typu B s fixovanými hraničními uzly

Podobně jako v předchozí části zde otestujeme obě volby monitorovacích funkcí. Při použití Winslowovy metody (3.43) pro  $\alpha = 1$ , velikost časového kroku umělého času  $\Delta\tau = 1$  a parametr ovlivňující rychlost adaptace sítě  $\mathcal{T} = 1$  při zvolených parametrech  $MAXITER = 1$ ,  $TSTEP = 20$  začal ukazatel dosažení stacionárního stavu  $REZ$  (3.71) prudce narůstat, což se mimo jiné projeví tak, že v důsledku nenalezení řešení není možné síť adaptovat. Toto chování bylo důvodem, proč jsme v dalším textu použili hodnotu  $\alpha = 0.1$ , resp.  $\alpha = 0.05$ <sup>24</sup>. Spočtené hodnoty jsou zaneseny v tabulce 3.7.

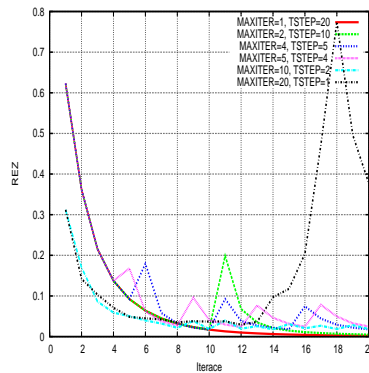
<sup>24</sup>Při použití vyššího počtu adaptačních kroků sítě (parametr  $MAXITER$ ) docházelo opět k nárůstu ukazatele  $REZ$  (3.71), byli jsme proto nuceni dále snížit hodnotu  $\alpha$  na  $\alpha = 0.05$ . Obdobný postup je použit i v dalších tabulkách. Toto skokové snížení hodnoty  $\alpha$  vede k tomu, že v obrázcích zachycujících průběh ukazatele  $REZ$  je v první iteraci tato hodnota různá, právě podle použitého  $\alpha$ , viz např. obrázek 3.27.



Obrázek 3.26: Úloha typu A. Adaptovaná síť při použití Winslowovy metody (vlevo) a monitorovací funkce ALLMF (vpravo),  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ ,  $MAXITER = 20$ ,  $TSTEP = 1$ .

$MAXITER$	$TSTEP$	$REZ$	$T[s]$	$\alpha$
1	20	0.002059	150.31	0.1
2	10	0.004881	151.34	0.1
4	5	0.017811	152.38	0.1
5	4	0.024871	155.19	0.1
10	2	0.018411	154.85	0.05
20	1	0.377652	166.59	0.05

Tabulka 3.7: Úloha typu B. Sledované veličiny při  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ , použita Winslowova metoda.



Obrázek 3.27: Úloha typu B. Průběh ukazatele REZ, při  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ , použita Winslowova metoda.

Z průběhu sledované veličiny  $REZ$ , na obrázku 3.27, je patrné, že při vyšším  $MAXITER$  dochází ke zhoršení tohoto ukazatele. Toto se prakticky může projevovat tak, že v adaptované síti dojde ke křížení některých hran sítě. Při realizaci algoritmu je na to brán zřetel a taková adaptace se neprovádí. Protože obdržené výsledky nejsou zcela optimální, byly provedeny testy pro různé kombinace parametrů  $\Delta\tau$  a parametru  $\mathcal{T}$ . Naměřené výsledky jsou shrnuty v následujících tabulkách 3.8 a 3.9. Průběhy sledovaných veličin pro naměřená data jsou zachyceny na obrázcích 3.28. Z nich je patrné, že při snížení hodnoty  $\Delta\tau = 1$  na  $\Delta\tau = 0.1$  se zásadně zmenšily oscilace veličiny  $REZ$  (3.71). Dále, že zvýšení parametru  $\mathcal{T} = 1$  na  $\mathcal{T} = 10$  nemá v tomto případě prakticky žádný vliv na vzniklé oscilace, vyjma toho, že číselná hodnota ukazatele  $REZ$  je nižší. Výsledné sítě pro  $MAXITER = 2$  a  $TSTEP = 10$  jsou na obrázku 3.29. Zde je vidět, že metoda částečně zachytila oblasti s největším gradientem. Opticky je nejhůře adaptovaná síť s největším  $\mathcal{T}$ .

$MAXITER$	$TSTEP$	$REZ$	$T[s]$	$\alpha$
1	20	0.025579	152.18	0.1
2	10	0.022592	152.37	0.1
4	5	0.022590	153.26	0.1
5	4	0.024353	154.41	0.1
10	2	0.013830	156.82	0.05
20	1	0.446084	160.79	0.05

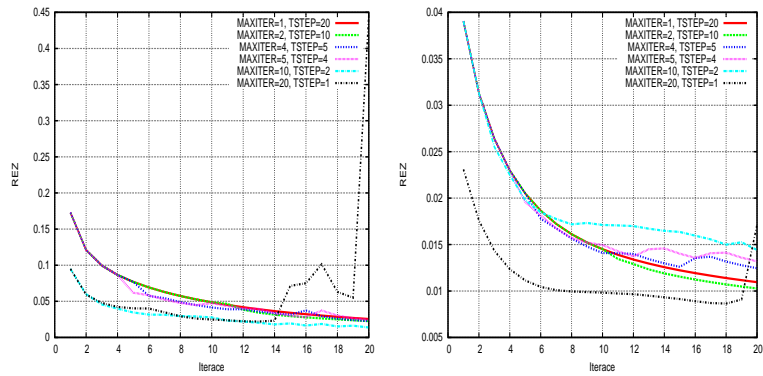
Tabulka 3.8: Úloha typu B. Sledované veličiny při  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$ , použita Winslowova metoda.

$MAXITER$	$TSTEP$	$REZ$	$T[s]$	$\alpha$
1	20	0.010948	151.31	0.1
2	10	0.010273	152.15	0.1
4	5	0.012448	153.10	0.1
5	4	0.013191	153.41	0.1
10	2	0.014423	156.62	0.1
20	1	0.017152	161.36	0.05

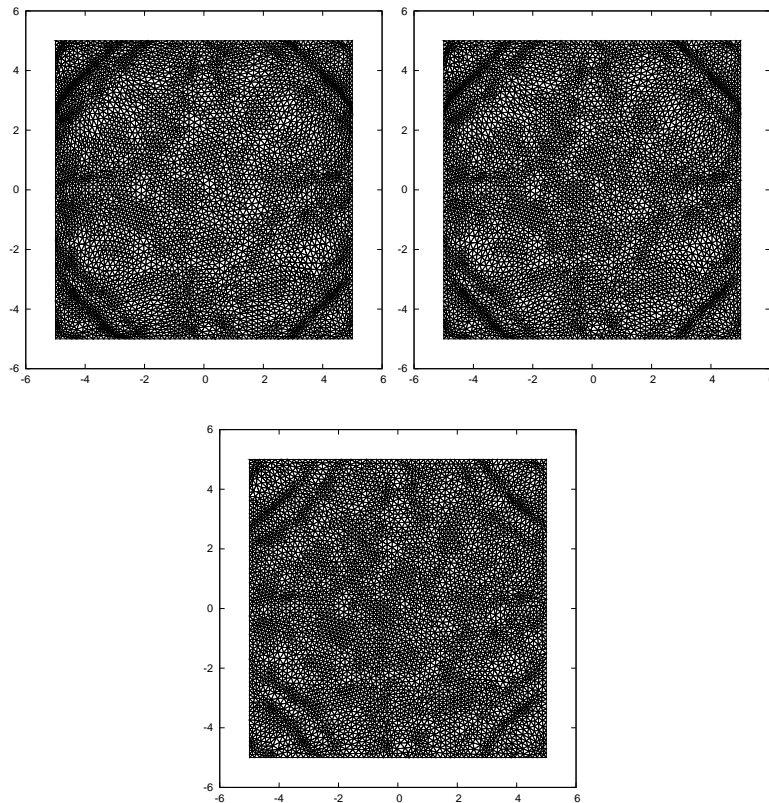
Tabulka 3.9: Úloha typu B. Sledované veličiny při  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$ , použita Winslowova metoda

Nyní otestujeme použití monitorovací funkce založené na harmonickém zobrazení (ALLMF), viz (3.47) a (3.49). V následující tabulce 3.10 jsou zachyceny získané hodnoty. Použité parametry jsou  $\Delta\tau = 1$  a  $\mathcal{T} = 1$ . Chování sledovaných veličin během adaptačního procesu je zachyceno na obrázku 3.30. Všimněme si, že při větším počtu kroků adaptace sítě, tedy pro vyšší hodnoty  $MAXITER$ , dochází k růstu  $REZ$  a navíc došlo při hodnotě parametru  $MAXITER = 10$  a  $MAXITER = 20$  k selhání výpočtů z důvodů dělení nulou při výpočtu koeficientů v (3.62). Je tedy vidět, že zde použití monitorovací funkce ALLMF nevedlo k uspokojivým výsledkům.

Opět jsme tedy jako v předchozím případě provedli další výpočty pro  $\Delta\tau = 0.1$  a  $\mathcal{T} = 1$ , resp.  $\mathcal{T} = 10$ . V následujících tabulkách 3.11 a 3.12 jsou zanesena



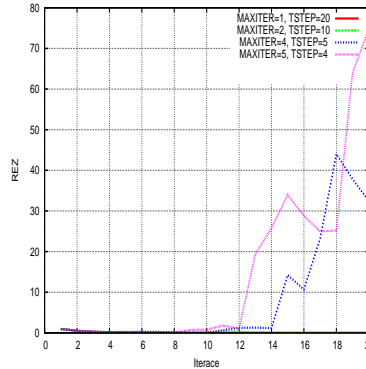
Obrázek 3.28: Úloha typu B. Průběh ukazatele REZ při  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$  (vlevo) a  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$  (vpravo), použita Winslowova metoda



Obrázek 3.29: Úloha typu B. Adaptovaná síť  $MAXITER = 2, TSTEP = 10$  pro parametry  $\Delta\tau = 1, \mathcal{T} = 1$ ;  $\Delta\tau = 0.1, \mathcal{T} = 1$ ;  $\Delta\tau = 0.1, \mathcal{T} = 10$ , použita Winslowova metoda.

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.001298	153.91
2	10	0.003815	154.65
4	5	32.45440	155.09
5	4	74.78550	154.23

Tabulka 3.10: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF.



Obrázek 3.30: Úloha typu B. Průběh ukazatele REZ pro  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF.

naměřená data. Opět, podobně jako v předchozím případě, docházelo k selhání výpočtu, z důvodu dělení nulou při výpočtu matic a koeficientů (3.62), konkrétně při výpočtu Jakobiánu  $\mathcal{J} = \frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_1}$ . V případech s vyšší hodnotou  $MAXITER$ , když výpočet proběhl, docházelo poměrně často k poruchám při adaptaci sítě, a tak daná adaptace nebyla provedena.

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.029396	152.23
2	10	0.024385	153.55
10	2	0.023504	154.10

Tabulka 3.11: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF.

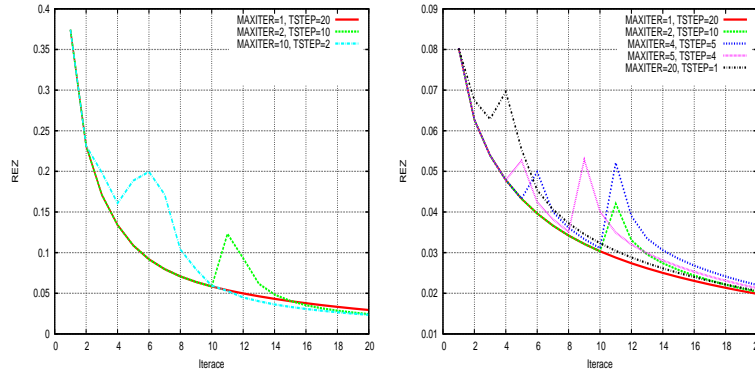
Průběhy ukazatele  $REZ$  (3.71) jsou zaneseny na obrázcích 3.31. Ze spočtených výsledků je vidět, že v případě monitorovací funkce ALLMF jsme ani v jedné z testovaných konfigurací nedospěli k uspokojivým výsledkům průběhu adaptačního procesu. Důvodem je to, že nejsme schopni korigovat vyšší hodnotu normy gradientu funkce  $u_B$ <sup>25</sup>, jejíž gradient používáme pro konstrukci monitorovací funkce  $\mathbb{G}$ . Poznamenejme, že pro funkci  $u_A$  v předchozích testech k tomuto jevu nedocházelo, protože maximální hodnota normy gradientu  $u_A$  byla nižší než u funkce  $u_B$ . Provedeme-li však optické hodnocení výsledných sítí, na obrázku 3.34 (vlevo), je vidět, že opticky jsou adaptovány uspokojivě.

Abychom výše uvedený nedostatek odstranili, použili jsme vlastní *ad-hoc* úpravu použité monitorovací funkce a to tak, že jednotlivé složky gradientu  $\nabla u_B$

<sup>25</sup>Na rozdíl od Winslowovy metody, kde tak činíme pomocí parametru  $\alpha$ .

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.019878	152.08
2	10	0.020344	152.70
4	5	0.022043	153.79
5	4	0.021277	153.90
20	1	0.020612	153.95

Tabulka 3.12: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$  za použití monitorovací funkce ALLMF.



Obrázek 3.31: Úloha typu B. Průběh ukazatele REZ pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$  (vlevo),  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$  (vpravo), za použití monitorovací funkce ALLMF.

byly znormovány pomocí jejich absolutní hodnoty, tj. místo hodnoty  $\nabla u$  ve vztahu (3.49) použijeme

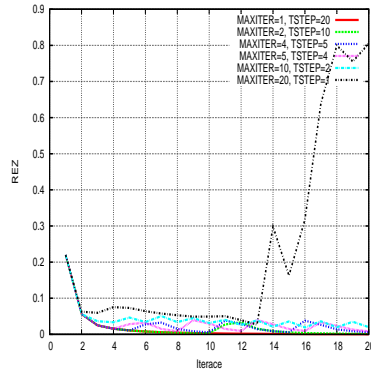
$$\overline{\nabla u} := \left( \frac{1}{mx_1} \frac{\partial u}{\partial x_1}, \frac{1}{mx_2} \frac{\partial u}{\partial x_2} \right)^T, \quad (3.81)$$

kde  $mx_1 = \max_{\mathbf{x} \in \Omega} \left( \left| \frac{\partial u}{\partial x_1} \right| \right)$  a  $mx_2 = \max_{\mathbf{x} \in \Omega} \left( \left| \frac{\partial u}{\partial x_2} \right| \right)$ <sup>26</sup>. Všimněme si, že při použití této náhrady je matice  $\mathbb{M}$  ve vztahu (3.49) symetrická a pozitivně definitní, a tedy opět dostáváme monitorovací funkci využívající harmonické zobrazení, viz vztah (3.47).

Následující tabulky 3.13, 3.14 a 3.15 shrnují naměřené výsledky pro takto upravenou monitorovací funkci. Zde je vidět, že ve všech případech byla doba potřebná pro adaptaci vyšší o cca 10%. Průběhy veličiny  $REZ$  jsou zaneseny na obrázcích 3.32, 3.33, kde je vidět, že normování ve většině případů zabránilo nárůstu ukazatele dosažení stacionárního stavu  $REZ$ . Porovnání výsledných sítí je zaneseno na obrázku 3.34, zde je vidět, že bez normování je síť opticky adaptována výrazně lépe, než s jeho použitím. Porovnáme-li sítě získané s použitím normování z obrázku 3.34 se sítěmi získanými pomocí Winslowovy metody na obrázku 3.29 pak je vidět, že získané sítě jsou obdobné. Z výše uvedeného vyplývá, že použití normování v případě monitorovací funkce ALLMF má stejný vliv jako snížení konstanty  $\alpha$  v případě Winslowovy metody.

<sup>26</sup>Poznamenejme, že příslušná maxima jsou určena z konečné prvkové reprezentace funkce  $u_B$ .





Obrázek 3.32: Úloha typu B. Průběh ukazatele REZ pro  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF s normováním.

<i>TSTEP</i>	<i>MAXITER</i>	<i>REZ</i>	<i>T[s]</i>
1	20	0.000413	163.74
2	10	0.001288	164.66
4	5	0.006090	165.63
5	4	0.008538	165.44
10	2	0.019962	168.00
20	1	0.804998	170.45

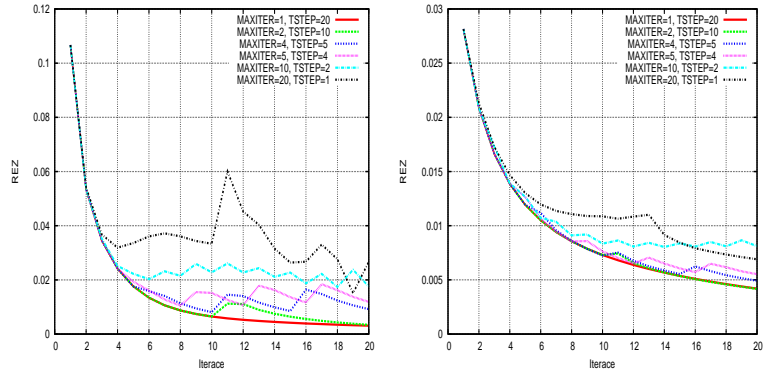
Tabulka 3.13: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF s normováním.

<i>TSTEP</i>	<i>MAXITER</i>	<i>REZ</i>	<i>T[s]</i>
1	20	0.003089	165.08
2	10	0.003418	166.66
4	5	0.009204	166.82
5	4	0.011876	167.72
10	2	0.017545	171.27
20	1	0.026683	174.04

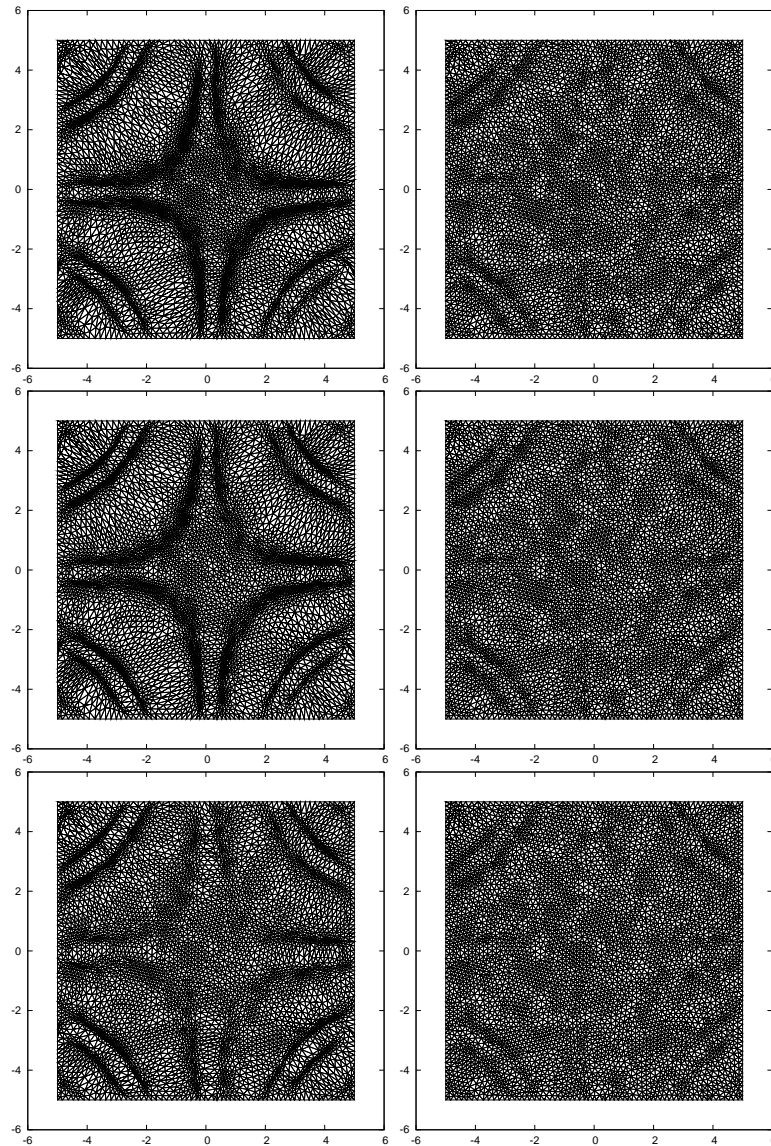
Tabulka 3.14: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF s normováním.

<i>TSTEP</i>	<i>MAXITER</i>	<i>REZ</i>	<i>T[s]</i>
1	20	0.004191	164.43
2	10	0.004167	165.47
4	5	0.004919	165.62
5	4	0.005522	168.07
10	2	0.008109	170.01
20	1	0.006902	170.47

Tabulka 3.15: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$  za použití monitorovací funkce ALLMF s normováním.



Obrázek 3.33: Úloha typu B. Průběh ukazatele REZ pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$  (vlevo),  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$  (vpravo), za použití monitorovací funkce ALLMF s normováním.



Obrázek 3.34: Úloha typu B. Adaptovaná síť  $MAXITER = 2$ ,  $TSTEP = 10$ , při použití monitorovací funkce ALLMF bez normování (vlevo) a s normováním (vpravo) pro  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ ;  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$ ;  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 10$ .

## Úloha typu A s posunem uzlů na hranici

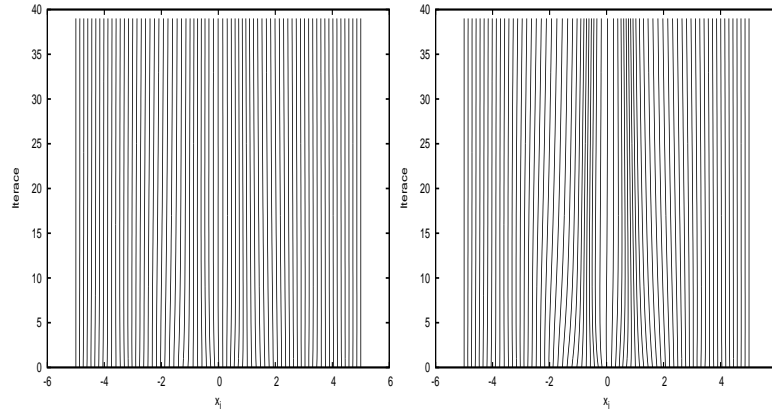
Použijme nyní stejné parametry jako v předchozí úloze, parametr rychlosti adaptace sítě  $\mathcal{T} = 1$ ,  $\Delta\tau = 1$ , kde  $\Delta\tau = \tau_{\nu+1} - \tau_\nu$  viz (3.64), při použití Winslowovy metody (3.43) volme  $\alpha = 1$ .

V dalším použijme následující značení:

$$\begin{aligned}\Gamma_T &= \{(x_1, x_2) \in \partial\Omega, \text{ kde } x_2 = 5\}, \\ \Gamma_B &= \{(x_1, x_2) \in \partial\Omega, \text{ kde } x_2 = -5\}, \\ \Gamma_L &= \{(x_1, x_2) \in \partial\Omega, \text{ kde } x_1 = -5\}, \\ \Gamma_R &= \{(x_1, x_2) \in \partial\Omega, \text{ kde } x_1 = 5\}.\end{aligned}$$

Volbu okrajových podmínek provedeme následovně, funkce  $u_A$  (3.79) je konstantní na  $\Gamma_T$  a  $\Gamma_B$ , proto zde předepíšeme pevné pozice uzlů. Části hranice  $\Gamma_L$  a  $\Gamma_R$  parametrizujeme<sup>27</sup>, viz (3.68), a řešíme zde 1D úlohu (3.53) pro parametr  $s$ , viz bod 3 algoritmu 3.1. Pro vlastní řešení 1D problému použijme metodu G-S iterací (3.58), která konverguje rychleji než metoda umělého času, v případě, že by došlo k oscilacím pohybu uzlů na hranici, můžeme použít metodu umělého času (3.54).

Z důvodů téměř zanedbatelného posunu uzlů na hranici, viz obrázek 3.35, použijeme v případě Winslowovy metody pro výpočet projekce  $M$  (3.69) monitorovací funkce  $\mathbb{G}$  (3.43) jiné hodnoty konstanty  $\alpha$  a označme ji  $\alpha_b$ ,  $\alpha_b = 10$ . Dále označme maximalní počet iterací schématu (3.58) pro posun hraničních uzlů jako  $MAXITER_b$  a volme  $MAXITER_b = 40$ . Hodnota minimálního posunu hraničních uzlů pro zastavení  $\epsilon_b$ , byla použita  $\epsilon_b = 10^{-6}$ , viz (3.77).



Obrázek 3.35: Úloha typu A. Trajektorie uzlů na levé hraně oblasti pro  $\alpha_b = 1$  (vlevo) a  $\alpha_b = 10$  (vpravo).

Výsledky pro různé kombinace parametrů  $MAXITER$  a  $TSTEP$  pro Winslowovu metodu a pro použití monitorovací funkce ALLMF ((3.47) a (3.49)) shrnoují následující tabulky<sup>28</sup> 3.16 a 3.17.

Průběhy sledovaných veličin se nijak zásadně neliší od průběhů pro stejnou testovací úlohu bez pohybu uzlů, obrázky 3.23 a proto je zde nebudeme uvádět. Ze

<sup>27</sup>Použili jsme parametrizace  $\psi_L(s) = B_L + (A_L - B_L)s$ , resp.  $\psi_R(s) = B_R + (A_R - B_R)s$ ,  $s \in [0, 1]$ , kde  $A_L = [-5, 5]$ ,  $B_L = [-5, -5]$ , resp.  $A_R = [5, 5]$ ,  $B_R = [5, -5]$ .

<sup>28</sup>Protože při použití malého počtu iteračních kroků  $TSTEP$  a většího počtu opakování adaptace  $MAXITER$  docházelo v předchozích experimentech ke zhoršování sledovaných parametrů, tak je nebereme v dalších experimentech v potaz.

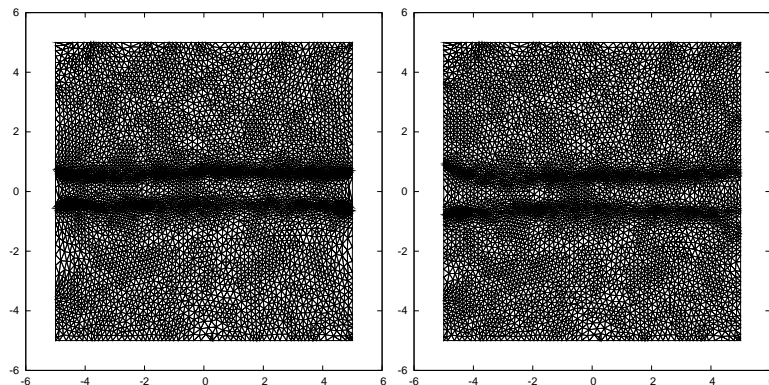
<i>MAXITER</i>	<i>TSTEP</i>	<i>REZ</i>	<i>T[s]</i>
1	20	0.001870	150.78
2	10	0.002664	151.13
4	5	0.004580	152.29
5	4	0.006620	153.96

Tabulka 3.16: Úloha typu A. Sledované veličiny při  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$ ,  $\alpha = 1$ ,  $\alpha_b = 10$  za použití Winslowovy metody.

<i>MAXITER</i>	<i>TSTEP</i>	<i>REZ</i>	<i>T[s]</i>
1	20	0.000750	152.79
2	10	0.001645	153.07
4	5	0.005418	154.08
5	4	0.004891	154.83

Tabulka 3.17: Úloha typu A. Sledované veličiny při  $\Delta\tau = 1$ ,  $\mathcal{T} = 1$  za použití monitorovací funkce ALLMF.

spočtených dat lze, při porovnání s úlohou bez pohybu uzlů, říci, že v tomto případě měl pohyb uzlů na hranici zanedbatelný vliv na sledované veličiny *REZ* (3.71) a *T*. Vliv pohybu krajních uzlů sítě na výslednou síť je zanesen na následujících obrázcích 3.36. Zde je vidět, že při použití monitorovací funkce ALLMF nedošlo téměř k posunu uzlů. Tento jev je způsoben tím, že v případě naší testovací funkce  $u_A$  platí, že  $\frac{\partial u_A}{\partial x_1} = 0$  a tedy monitorovací funkce daná (3.49) je při použití projekce (3.69) na levou a pravou hranici oblasti totožná s Winslowovou metodou s  $\alpha = 1$ , pro kterou nedošlo téměř k žádnému posunu uzlů, viz obrázek 3.35. V případě Winslowovy metody je vidět, že k posunu uzlů došlo, ale hustota uzlů na levé a pravé hranici je jiná, než hustota uzlů uvnitř oblasti. Tento jev je dán tím, že jsme použili různou volbu parametru  $\alpha$  v monitorovací funkci  $\mathbb{G}$ , viz (3.43) a (3.44), uvnitř oblasti  $\Omega$  ( $\alpha = 1$ ) a na jejích hranicích ( $\alpha_b = 10$ ), a tyto parametry mají vliv na výslednou adaptaci, viz 1D experimenty kapitola 3.1.6.



Obrázek 3.36: Úloha typu A. Výsledné sítě pro Winslowovu metodu (vlevo) a monitorovací funkci ALLMF (vpravo), použité parametry  $MAXSTEP = 5$ ,  $TSTEP = 4$ .

## Úloha typu B s posunem uzlů na hranici

Nyní otestujme posun uzlů na hranici pro úlohu s funkcí  $u_B$ . Na základě provedeného experimentu s pevnými uzly na hranici volme následující parametry výpočtu: konstantu  $\mathcal{T} = 1$ ,  $\Delta\tau = 0.1$ , pro Winslowovu metodu (3.43) volme  $\alpha = 0.1$  a stejnou hodnotu volme i pro hranici, tedy  $\alpha_b = 0.1$ . V případě monitorovací funkce ALLMF ((3.47) a (3.49)), volme monitorovací funkci s normováním.

V tomto případě dochází k pohybu uzlů po všech čtyřech částech hranice  $\Gamma_L$ ,  $\Gamma_R$ ,  $\Gamma_T$  a  $\Gamma_B$ . Pohyb uzlů na hranici je opět řešen pomocí G-S iterací (3.58), obdobně jako v předchozí úloze s funkcí  $u_A$ . Maximalní počet iterací pro posun hraničních uzlů volme  $MAXITER_b = 40$  a hodnota minimálního posunu hraničních uzlů pro zastavení  $\epsilon_b$ , viz (3.77), byla použita  $\epsilon_b = 10^{-6}$ .

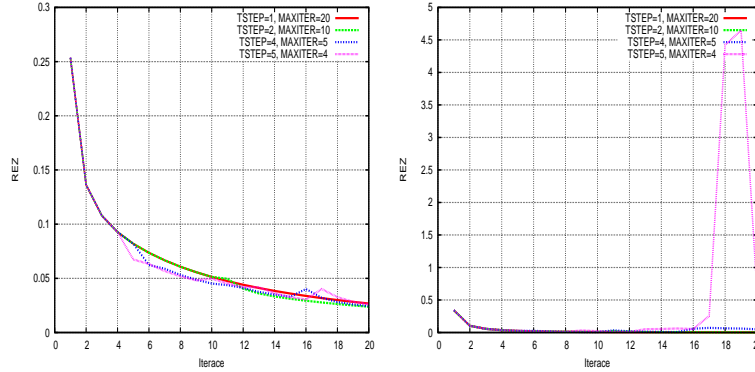
Získané výsledky jsou v tabulkách 3.18 a 3.19. Obrázky 3.37 zachycují průběh sledovaných veličin, opět zde v případě použití monitorovací funkce ALLMF došlo k nárůstu ukazatele  $REZ$ , čili se vyskytl problém s konvergencí metody. Na obrázku 3.38 jsou zaneseny výsledné sítě, zde je vidět, že z hlediska optického hodnocení jsou výsledné sítě obdobné.

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.026810	150.48
2	10	0.023806	151.59
4	5	0.024340	152.4
5	4	0.026065	152.65

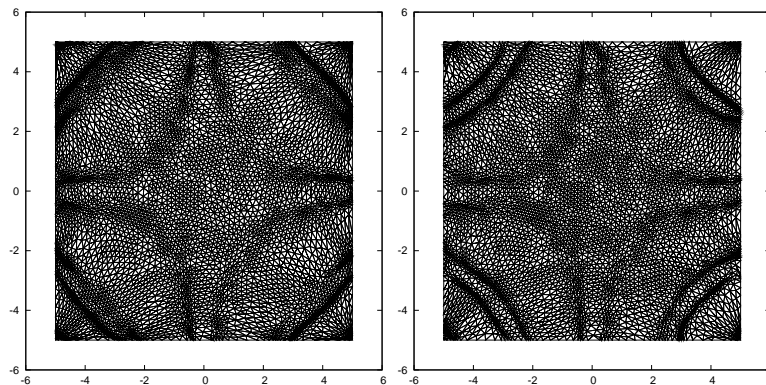
Tabulka 3.18: Úloha typu B. Sledované veličiny pro  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$ ,  $\alpha = 0.1$ ,  $\alpha_b = 0.1$ , použita Winslowova metoda.

$MAXITER$	$TSTEP$	$REZ$	$T[s]$
1	20	0.003319	164.97
2	10	0.002538	164.42
4	5	0.049952	165.64
5	4	0.611225	166.07

Tabulka 3.19: Úloha typu B. Sledované veličiny při  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$ , použita monitorovací funkce ALLMF s normováním.



Obrázek 3.37: Úloha typu B. Průběh ukazatele REZ,  $\Delta\tau = 0.1$ ,  $\mathcal{T} = 1$ , Winslowova metoda (vlevo), monitorovací funkce ALLMF s normováním (vpravo).



Obrázek 3.38: Úloha typu B. Výsledné sítě pro Winslowovu metodu (vlevo) a monitorovací funkci ALLMF s normováním (vpravo), použité parametry  $MAXSTEP = 5$ ,  $TSTEP = 4$ .

## Zhodnocení provedených experimentů ve 2D

Z provedených experimentů vyplývá, že metoda je schopna zachytit oblasti s vyšší hodnotou normy gradientu, či nespojitostí. Metoda je citlivá na volbu časového kroku umělého času  $\Delta\tau$ . Při jeho nevhodné volbě docházelo, buď k překřížení hran sítě, nebo ke zhroucení výpočtu, což je programově ošetřeno tak, že než je provedena adaptace sítě, krok 7. algoritmu 3.1, tak je otestováno pomocí nástrojů FreeFem++, zda nedojde k překřížení hran.

Dále se při několikanásobné adaptaci ukázalo, že ukazatel  $REZ$  se v průběhu výpočtu, po provedení adaptace, dle algoritmu 3.1 sice může zvýšit, ale při “vhodně” zvolených parametrech dochází v dalších krocích k jeho poklesu, pokud ne, dojde ke zhroucení výpočtu, viz provedené experimenty.

Větší možnost ovlivnit průběh adaptace nám nabízí díky parametru  $\alpha$  Winslowova metoda, a z toho důvodu ji budeme používat v dalších numerických experimentech.

Pro zvýšení použitelnosti námi užitého algoritmu by bylo do budoucna zajímavé studovat problematiku stabilizace tohoto výpočetního schématu.

## 3.2 Anisotropní adaptivita

V této kapitole se budeme krátce zabývat anisotropní adaptací sítě (AMA-Anisotropic Mesh Adaptation) v 1D a 2D, viz např. práce V. Dolejšího [10], či popis uvedený v [13], jež je základem pro námi vyvinuté algoritmy.

Označme jako  $\mathbf{w}_h^*$  lineární rekonstrukci získanou z po částech konstantního numerického řešení  $\mathbf{w}_h^k$  Eulerových rovnic (1.17) v čase  $t_k^{29}$ , viz definice 2.1. Pro získání výše zmíněné lineární rekonstrukce budeme používat vztahu:

$$\mathbf{w}_h^*(P_j) = \frac{\sum_{D_i \in \mathcal{K}_{P_j}} |D_i| \mathbf{w}_i^k}{\sum_{D_i \in \mathcal{K}_{P_j}} |D_i|}, \quad (3.82)$$

kde  $\mathbf{w}_h^*(P_j)$  je hodnota rekonstrukce  $\mathbf{w}_h^*$  ve vrcholu  $P_j$ ,  $\mathbf{w}_i^k$  je numerické řešení na objemu  $D_i$  v čase  $t_k$  a  $\mathcal{K}_{P_j}$  označuje množinu objemů, které mají společný vrchol  $P_j$ . Jako  $u_h^* \in \mathcal{L}_{\mathcal{T}_h}$  označme po částech lineární fyzikální veličinu zkonstruovanou z  $\mathbf{w}_h^*$ , kde  $\mathcal{L}_{\mathcal{T}_h}$  je prostor po částech lineárních funkcí na triangulaci  $\mathcal{T}_h$ . Dále si definujme chybu diskretizace  $e_h$ :

$$e_h := \|u - u_h^*\|_W, \quad (3.83)$$

kde funkcí  $u \in W$  rozumíme fyzikální veličinu danou přesným řešením  $\mathbf{w}(\cdot, t_k)$  Eulerových rovnic (1.17) v čase  $t_k$ ,  $W$  je vhodný prostor funkcí a  $\|\cdot\|_W$  je vhodná norma. Naším cílem je udržet chybu diskretizace pro námi spočtené řešení pod předepsanou tolerancí  $TOL$ , tedy musíme splnit následující podmínku:

$$e_h \leq TOL \quad (3.84)$$

Dále si definujme *operátor nejlepší aproximace*  $\pi_h : W \rightarrow \mathcal{L}_{\mathcal{T}_h}$  jako:

$$\|u - \pi_h u\|_W := \min_{v \in \mathcal{L}_{\mathcal{T}_h}} \|u - v\|_W. \quad (3.85)$$

Nutnou podmínkou pro splnění (3.84) je splnění této podmínky<sup>30</sup>:

$$\|u - \pi_h u\|_W \leq TOL. \quad (3.86)$$

V dalším předpokládejme, že  $u \in C^1(\bar{\Omega})$  a definujme *interpolační operátor*  $r_h : C^1(\bar{\Omega}) \rightarrow \mathcal{L}_{\mathcal{T}_h}$  tak, aby pro všechny objemy  $D_i \in \mathcal{T}_h$  splňoval následující dvě podmínky:

$$\begin{aligned} r_h u(\mathbf{x}_{D_i}) &= u(\mathbf{x}_{D_i}), \\ \nabla r_h u(\mathbf{x}_{D_i}) &= \nabla u(\mathbf{x}_{D_i}), \end{aligned}$$

kde  $\mathbf{x}_{D_i} \in D_i$  je pevně určený bod v objemu  $D_i$ . Z definice operátorů  $r_h$  a  $\pi_h$  je zřejmé, že podmínka (3.86) je splněna, když je splněno následující:

$$\|u - r_h u\|_W \leq TOL. \quad (3.87)$$

<sup>29</sup>Z důvodů přehlednosti nebudeme u  $\mathbf{w}_h^*$  zavádět horní index  $k$  označující čas  $t_k$ .

<sup>30</sup>Poznamenejme, že, aby numerická metoda pro řešení (1.17) měla vůbec šanci splnit podmínku (3.84), je nutné mít síť  $\mathcal{T}_h$ , na které je splněna podmínka (3.86) [13]. V dalším se budeme konstruovat právě takové síť zabývat.



Bude-li splněn požadavek, že  $u \in C^2(\overline{\Omega})$ , lze pomocí Taylorova rozvoje v bodě  $\mathbf{x}_{D_i} \in D_i$  psát

$$u(\mathbf{x}) - r_h u(\mathbf{x}) = \frac{1}{2} (\mathbf{x} - \mathbf{x}_{D_i})^T \mathbb{H}(\mathbf{x}_{D_i}) (\mathbf{x} - \mathbf{x}_{D_i}) + o(|\mathbf{x} - \mathbf{x}_{D_i}|^2), \quad (3.88)$$

kde

$$\mathbb{H}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 u}{\partial x_1^2} & \frac{\partial^2 u}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 u}{\partial x_1 \partial x_N} \\ \frac{\partial^2 u}{\partial x_2 \partial x_1} & \frac{\partial^2 u}{\partial x_2^2} & \cdots & \frac{\partial^2 u}{\partial x_2 \partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 u}{\partial x_N \partial x_1} & \frac{\partial^2 u}{\partial x_N \partial x_2} & \cdots & \frac{\partial^2 u}{\partial x_N^2} \end{pmatrix} (\mathbf{x}), \quad (3.89)$$

je Hessova matice. Levá část výrazu (3.88) představuje chybu interpolace, definujme proto funkci  $I_u(x)$  popisující chybu interpolace<sup>31</sup> následovně:

$$I_u(x) := |u(x) - r_h u(x)|. \quad (3.90)$$

V dalším bude naším cílem adaptovat výpočetní síť takovým způsobem, aby byla chyba interpolace  $I_u(x)$  ekvidistribuována po výpočetní oblasti  $\Omega$ , tedy  $I_u(x) \approx C \forall x \in \Omega$ . Chybu interpolace na hraně  $e$  lze aproximovat pomocí střední hodnoty  $I_u$  na této hraně a při zanedbání členů vyšších řádů ve vztahu (3.88) obdržíme [13]:

$$I_u|_e \approx \frac{1}{|e|} \int_e |u(\mathbf{x}) - r_h u(\mathbf{x})| dS \approx \frac{1}{24} |\mathbf{e}^T \mathbb{H}(\mathbf{x}_e) \mathbf{e}|, \quad (3.91)$$

kde  $\mathbf{e}$  je vektor tvořící hranu  $e$ , bod  $\mathbf{x}_e$  označuje její střed a  $|e|$  je její délka. Za předpokladu pozitivní definitnosti Hessovy matice  $\mathbb{H}(\mathbf{x}_e)$  lze zavést *Riemannovu normu* hrany  $e$ :

$$\|e\|_u = (\mathbf{e}^T \mathbb{H}(\mathbf{x}_e) \mathbf{e})^{1/2}. \quad (3.92)$$

Požadavek na ekvidistribuci chyby interpolace lze pak psát pomocí Riemannovy normy ve tvaru:

$$I_u|_e \approx \frac{1}{24} \|e\|_u^2 \approx c, \quad (3.93)$$

po všechny hrany sítě  $\mathcal{T}_h$ . Toto nás vede k tomu, že síť  $\mathcal{T}_h$  je optimální, jestliže pro všechny hrany sítě platí:

$$\|e\|_u \approx c_N, \quad (3.94)$$

kde  $N$  je dimenze úlohy a  $c_N$  je konstanta závislá na dimenzi úlohy, viz dále. Pro zhodnocení optimality sítě definujme parametr kvality sítě  $Q_{\mathcal{T}_h}$ , jenž nám udává kvalitu sítě ve smyslu nejmenších čtverců, tedy:

$$Q_{\mathcal{T}_h} = \frac{1}{J} \sum_{D_i \in \mathcal{T}_h} \sum_{e=\text{hrana } D_i} (\|e\|_u - c_N)^2, \quad (3.95)$$

kde  $J$  je počet objemů sítě.

Hodnoty  $c_N$  je možné určit z geometrické interpretace Riemannovy normy. Ze vztahu (3.88) je, při předpokladu pozitivní definitnosti Hessovy matice  $\mathbb{H}(\mathbf{x}_{D_i})$

---

<sup>31</sup>Interpolation error function

a zanedbání členů vyšších řádů, vidět, že množina všech bodů, které mají chybu interpolace menší než je předepsaná tolerance  $TOL$ , je elipsa ( $N = 2$ ). Naší snahou je vepsat do této množiny objem (pro  $N = 2$  trojúhelník) s maximální<sup>32</sup>  $N$ -dimensionální mírou. Pro tento objem platí, že všechny jeho hrany jsou ekvilaterální v Riemannově normě a tedy hodnoty  $c_N$  jsou dány [13]:

$$c_N := \begin{cases} 2\sqrt{2TOL}, & N = 1, \\ \sqrt{3}\sqrt{2TOL}, & N = 2. \end{cases} \quad (3.96)$$

### 3.2.1 Konstrukce aproximace Hessovy matice

Protože při vlastním výpočtu máme k dispozici pouze hodnotu  $u_h^* \in \mathcal{L}_{\mathcal{T}_h}$  v každém z vrcholů sítě a nikoliv přesné řešení  $u$ , určíme aproximaci druhé derivace  $\partial^2 u_h^* / \partial x_k \partial x_l$  v Hessově matici pomocí funkce  $u_h^{*x_k x_l} \in \mathcal{L}_{\mathcal{T}_h}$ . V dalším předpokládejme<sup>33</sup>, že dimenze  $N = 2$  a označme si množinu všech uzlů triangulace  $\mathcal{T}_h$  jako  $\sigma_h$ . Použijme Greenovu větu a hledejme aproximaci  $u_h^{*x_k x_l}$  v následujícím slabém smyslu:

$$\left( \frac{\partial^2 u_h^*}{\partial x_k \partial x_l}, \varphi \right) \approx (u_h^{*x_k x_l}, \varphi) = \int_{\partial\Omega} \frac{\partial u_h^*}{\partial x_l} \varphi n_k dS - \left( \frac{\partial u_h^*}{\partial x_l}, \frac{\partial \varphi}{\partial x_k} \right) \quad \forall \varphi \in \mathcal{L}_{\mathcal{T}_h}, \quad (3.97)$$

kde  $(\cdot, \cdot)$  rozumíme skalární součin na  $L^2(\Omega)$ . Díky definici  $u_h^*$  a  $\varphi$  jsou jejich parciální derivace definované a tedy integrály na pravé straně (3.97) mají smysl. Pomocí numerické integrace lze tyto integrály snadno vyčíslit<sup>34</sup>. Zvolíme-li jako testovací funkce následující množinu funkcí:

$$\{\varphi_i; \varphi_i \in \mathcal{L}_{\mathcal{T}_h}, \varphi_i(P_j) = \delta_{ij}, i, j \in I\}, \quad (3.98)$$

kde  $P_j$  je některý z vrcholů sítě,  $\delta_{ij}$  je Kroneckerovo delta a  $I$  je indexová množina uzlů sítě, tak pro levou stranu lze za použití numerické integrace psát:

$$\begin{aligned} (u_h^{*x_k x_l}, \varphi_i) &= \sum_{D \in \mathcal{T}_h} \int_D u_h^{*x_k x_l} \varphi_i dx \\ &\approx \sum_{D \in \mathcal{T}_h} \frac{|D|}{N+1} \sum_{P_j \in D \cap \sigma_h} u_h^{*x_k x_l}(P_j) \varphi_i(P_j) \\ &= \left( \sum_{D \cap \sigma_h \ni P_i} \frac{|D|}{N+1} \right) u_h^{*x_k x_l}(P_i), \end{aligned} \quad (3.99)$$

kde  $N$  je dimenze. První z integrálů pravé strany (3.97) je vyhodnocen následovně:

$$\int_{\partial\Omega} \frac{\partial u_h^*}{\partial x_l} \varphi_i n_k dS = \sum_{j \in I_b} \frac{\partial u_h^*}{\partial x_l} \Big|_{S_j} n_k^j \int_{S_j} \varphi_i dS = \frac{1}{N} \sum_{S_j \ni P_i} |S_j| \frac{\partial u_h^*}{\partial x_l} \Big|_{S_j} n_k^j, \quad (3.100)$$

kde  $I_b$  značí indexní množinu všech hraničních hran  $S_j$  a symbol  $n_k^j$  značí  $k$ -tou složku vnější normály k této hraně. Dále jsme použili pro určení integrálu

<sup>32</sup>Vede to k tomu, že je objemů minimální počet.

<sup>33</sup>Obdobně bychom mohli postupovat i pro případ  $N = 1$ , kdy nahradíme aproximaci Hessovy matice její 1D analogií  $u_h^{*xx}(P_i)$ , tento postup je proveden v kapitole 3.3.4, vztah (3.130).

<sup>34</sup>parciální derivace funkcí  $u_h^*$  a  $\varphi$  jsou konstanty

v druhém členu vlastností (3.98) funkce  $\varphi_i$  a tedy sumace v posledním členu je prováděna přes všechny hraniční hrany, které mají společný vrchol  $P_i$ .

Druhý z integrálů pravé strany (3.97) je vyhodnocen:

$$\left( \frac{\partial u_h^*}{\partial x_l}, \frac{\partial \varphi_i}{\partial x_k} \right) = \sum_{D \ni P_i} |D| \left. \frac{\partial u_h^*}{\partial x_l} \right|_D \left. \frac{\partial \varphi_i}{\partial x_k} \right|_D. \quad (3.101)$$

Dosazením (3.99), (3.100) a (3.101) do (3.97) dostáváme vztah pro  $u_h^{*x_k x_l}$  v každém z vrcholů sítě  $P_i$ :

$$u_h^{*x_k x_l}(P_i) = \frac{\chi}{N} \sum_{S_j \ni P_i} |S_j| \left. \frac{\partial u_h^*}{\partial x_l} \right|_{S_j} n_k^j - \chi \sum_{D \ni P_i} |D| \left. \frac{\partial u_h^*}{\partial x_l} \right|_D \left. \frac{\partial \varphi_i}{\partial x_k} \right|_D, \quad (3.102)$$

$$\chi = \frac{N+1}{\sum_{D \cap \sigma_h \ni P_i} |D|}.$$

Pro určení Hessovy matice na hraně  $e$  použijeme následující aproximaci:

$$\mathbb{H} \left( \frac{P_i + P_j}{2} \right) \approx (\mathbb{H}(P_i) + \mathbb{H}(P_j)) / 2, \quad (3.103)$$

kde  $P_i$  a  $P_j$  jsou koncové uzly hrany  $e$ . Poznamenejme, že aproximace Hessovy matice zkonstruovaná tímto způsobem je symetrická [13, 10].

Protože není zaručena pozitivní definitnost Hessovy matice  $\mathbb{H}$ , kterou jsme předpokládali, použijeme místo ní pozitivně definitní matici  $\mathbb{M}$ , která nám zároveň dovolí kontrolovat počet objemů sítě. Uvažujme nejprve následující dekompozici matice  $\mathbb{H}$ :

$$\mathbb{H}(P) = \mathbb{R} \text{diag}(\lambda_1, \dots, \lambda_N) \mathbb{R}^{-1}, \quad (3.104)$$

kde  $\mathbb{R}$  je ortogonální matice a  $\lambda$  jsou vlastní čísla  $\mathbb{H}$  a zkonstruujeme pozitivně definitní matici  $\tilde{\mathbb{H}}$ :

$$\tilde{\mathbb{H}}(P) := \mathbb{R} \text{diag}(|\lambda_1|, \dots, |\lambda_N|) \mathbb{R}^{-1}. \quad (3.105)$$

Modifikovaná matice  $\mathbb{M}$  je pak definována následovně:

$$\mathbb{M}(P) := \bar{c} \left[ \mathbb{I} + \bar{\alpha}(\|\tilde{\mathbb{H}}(P)\|) \tilde{\mathbb{H}}(P) \right], \quad (3.106)$$

kde  $\mathbb{I}$  je jednotková matice  $N \times N$ ,  $\bar{c} > 0$  je konstanta a  $\bar{\alpha}$  je funkce, která je definována následovně:

$$\bar{\alpha}(\|\tilde{\mathbb{H}}(P)\|) := \frac{\varepsilon_1}{\varepsilon_2 + \|\tilde{\mathbb{H}}(P)\|}, \quad (3.107)$$

kde  $\|\tilde{\mathbb{H}}(P)\| = \max_{i,j=1,\dots,N} |\tilde{h}_{i,j}|$ ,  $\tilde{h}_{i,j}$  jsou prvky matice  $\tilde{\mathbb{H}}$ . A  $\varepsilon_1, \varepsilon_2$  jsou konstanty, jejichž poměr  $p = \varepsilon_1/\varepsilon_2$  ovlivňuje přechod mezi hrubou a zjemněnou částí sítě. Poznamenejme, že konstanta  $\varepsilon_1$  má následující význam:

$$\varepsilon_1 = \left( \frac{l_{max}}{l_{min}} \right)^2 - 1, \quad (3.108)$$

kde  $l_{max}$ , resp.  $l_{min}$  představují nejdelší, resp. nejkratší dosažitelnou délku hrany pro optimální síť v Euklidově metrice. Hodnotu konstanty  $\bar{c}$  lze na základě úvah uvedených v [13] určit pro  $N = 1, 2$

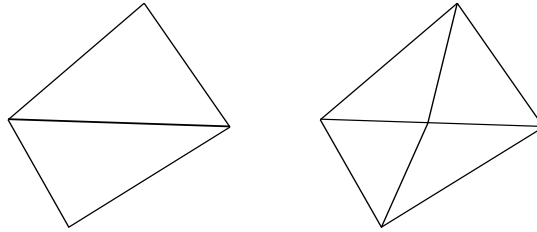
$$\bar{c} := \begin{cases} \left( \frac{J^{new} c_1}{|\Omega|} \right)^2 & N = 1, \\ \frac{\sqrt{3} J^{new} c_2^2}{4|\Omega|}, & N = 2, \end{cases} \quad (3.109)$$

kde  $c_1$  a  $c_2$  jsou dány pomocí vztahu (3.96) a  $J^{new}$  je požadovaný počet objemů na nové síti.

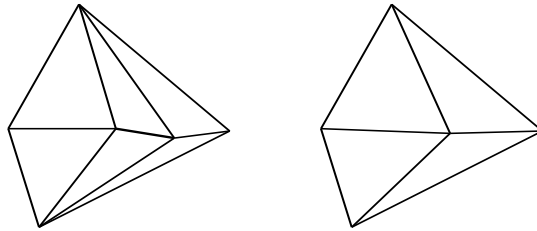
### 3.2.2 Minimalizace parametru kvality sítě

Pro minimalizaci kvality parametru sítě (3.95) 1D a ve 2D se v práci [10, 9] používají operace bisekce, odstranění hrany<sup>35</sup>, přehození hrany<sup>36</sup> a posun uzlu.

Na následujících obrázcích jsou schematicky znázorněny výše uvedené operace ve 2D. Poznamenejme, že při realizaci všech uvedených operací je třeba vzít v úvahu, zda se jedná o hranu hraniční (má oba své uzly na hranici), nebo hranu která má na hranici pouze jeden, či žádný uzel. Obdobně je tomu i v případě pohybu uzlů. Pro detailnější popis těchto operací je možno odkázat na [10, 9].



Obrázek 3.39: Bisekce naznačené hrany.



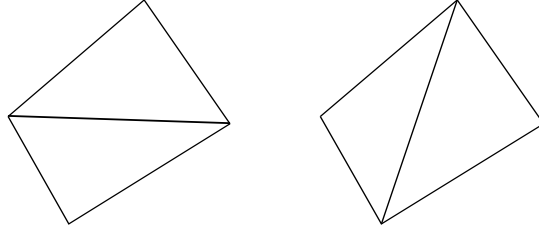
Obrázek 3.40: Odstranění naznačené hrany.

O tom, kdy se tyto operace provádějí, se rozhoduje pomocí Riemannovy normy dané hrany  $e$ :

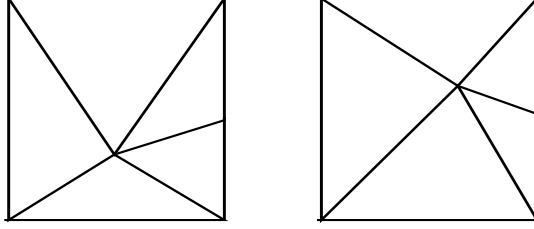
- Bisekce hrany ( $\mathcal{B}$ ) :  $\|e\|_u > \frac{4}{3}c_N$
- Odstranění hrany ( $\mathcal{R}$ ) :  $\|e\|_u < \frac{2}{3}c_N$
- Přehození hrany ( $\mathcal{S}$ ), operace je vykonávána pouze, když vede ke snížení parametru kvality sítě  $Q_{\mathcal{T}_h}$ .

<sup>35</sup>V 1D jde pouze o odstranění daného objemu.

<sup>36</sup>V 1D se neuskutečňuje.



Obrázek 3.41: Přehození naznačené hrany.



Obrázek 3.42: Posun společného uzlu.

Pro pohyb uzlů sítě ( $\mathcal{M}$ ) je v [13] použita lokální minimalizace parametru kvality pomocí metody největšího spádu, další strategie je uvedena v [10]. Výše uvedené operace pro modifikaci sítě se provádějí v určité posloupnosti, v [9] je uvedeno následující schéma:

$$m \times [\mathcal{M} + n \times [\mathcal{R} + \mathcal{S}] + \mathcal{M} + n [\mathcal{B} + \mathcal{S}] + \mathcal{M} + \mathcal{S}] + \mathcal{M}, \quad (3.110)$$

kde  $\mathcal{B}$ ,  $\mathcal{S}$ ,  $\mathcal{R}$ , představují bisekci, přehození, odstranění hran sítě a  $\mathcal{M}$  je pohyb uzlů sítě. Operace se provádějí nad celou sítí v uvedeném pořadí maximálně  $m$  krát, kde  $m = 10$ , symbolem  $n \times []$  rozumíme, že operace uvedené v hranatých závorkách se provádějí tak dlouho, dokud dochází ke změně sítě. Poznamenejme, že všechny výše uvedené 2D operace se provádějí pouze v případě, že nevznikne objem s malými úhly. Toto zajišťuje následující podmínka<sup>37</sup>:

$$\sum_{e=\text{hrana } D} |e|^N \leq \delta_{\min} |D|, \quad D \in \mathcal{T}_h, \quad (3.111)$$

kde  $\delta_{\min}$  je uživatelem definovaná konstanta ( $\delta_{\min} = 0.002$ ) a  $N = 2$ .

---

<sup>37</sup>Minimal angle condition

### 3.3 Metody založené na minimalizaci parametru kvality vrcholu

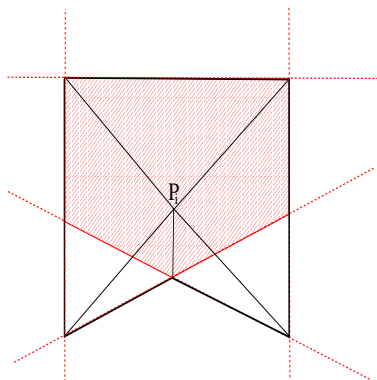
V této části práce si popíšeme **námi navržené** metody pro pohyb uzlů sítě [16, 17, 18, 21], jež jsou založeny na anisotropní adaptivitě. Pro vlastní adaptaci sítě pomocí anisotropní adaptace existuje celá řada programů, např. Angener verze 3.1<sup>38</sup>. Původnost navržené metody spočívá v minimalizaci parametru kvality vrcholu  $Q_{P_i}$  (3.114) za použití pouze pohybu uzlů za pomoci různých optimalizačních strategií, které, jak uvidíme později, ovlivňují chování celého algoritmu, a navíc používají specifický způsob interpolace hodnot numerického řešení na adaptované síti, viz kapitola 4. Metody jsou navrženy jak pro 1D, tak i pro 2D.

Abychom adaptovali síť, tj. zmenšili parametr kvality sítě  $Q_{\mathcal{T}_h}$  viz (3.95), použijme pouze pohyb jednotlivých vrcholů  $P_i$  triangulace  $\mathcal{T}_h$ . V dalším se zaměříme na 2D úlohy, čili úlohy kde  $N = 2$ . Buď  $P_i \in \sigma_h$  vrchol sítě  $\mathcal{T}_h$ , kde  $\sigma_h$  je množina všech vrcholů sítě, jako  $\mathcal{K}_{P_i}$  si označme oblast (viz obrázek 3.43) :

$$\mathcal{K}_{P_i} := \bigcup_{\substack{D_j \in \mathcal{T}_h \\ D_j \ni P_i}} D_j. \quad (3.112)$$

Pro každou hranu sítě  $e_j \in \partial\mathcal{K}_{P_i}$  uvažujme polorovinu  $\rho_j$ , pro jejíž hraniční přímku  $p_j$  platí  $p_j|_{\partial\mathcal{K}_{P_i}} = e_j$  a  $P_i \in \rho_j$ , množinu všech těchto polorovin příslušných k vrcholu  $P_i$  označme  $\Pi_{P_i}$ . Definujme přípustnou oblast pro pohyb vrcholu  $P_i$  jako:

$$\mathcal{K}_{P_i}^{ad} := \bigcap_{\rho_j \in \Pi_{P_i}} \rho_j. \quad (3.113)$$



Obrázek 3.43: Na obrázku je tučně naznačena hranice  $\partial\mathcal{K}_{P_i}$ . Červeně jsou naznačeny hraniční přímky  $p_j$  a červeným šrafováním je označena oblast  $\mathcal{K}_{P_i}^{ad}$ .

Množinu všech hran  $e$  sítě, které spojují vrchol  $P_i$  s vrcholem  $P_j \in \partial\mathcal{K}_{P_i} \cap \sigma_h$ , označme  $\mathcal{E}_{P_i}$ , jedná se tedy o množinu všech hran sítě, které obsahují vrchol  $P_i$ . Na základě ((3.94) a (3.95)) definujme pro vrchol  $P_i$  parametr kvality vrcholu  $Q_{P_i}$ :

$$Q_{P_i}(\mathbf{x}) := \sum_{e \in \mathcal{E}_{P_i}} (||e||_u - c_N)^2, \quad (3.114)$$

<sup>38</sup>Dostupné na <http://www.karlin.mff.cuni.cz/dolejsi/angen/angen3.1.htm>

kde  $\mathbf{x} = (x_1, \dots, x_N)^T$  je poloha vrcholu  $P_i$ . Tento parametr kvality vrcholu budeme minimalizovat tak, že budeme tímto vrcholem pohybovat. Tyto jednotlivé minimalizace budeme postupně cyklicky opakovat pro všechny vrcholy sítě  $\mathcal{T}_h$ . Při každém pohybu vrcholu  $P_i$  je nutné zaručit, že nedojde k překřížení hran, tj. vrchol bude po přemístění v přípustné oblasti  $\mathcal{K}_{P_i}^{ad}$ . Pro pohyb hraničních vrcholů sítě je ovšem třeba ještě vzít v úvahu to, že i po jejich posunutí se musí stále nacházet na hranici.

### 3.3.1 Metoda používající bariérovou funkci

Abychom zajistili, že nehraniční vrchol  $P_i$  zůstane při minimalizaci v oblasti  $\mathcal{K}_{P_i}^{ad}$ , budeme používat naši původní metodu vycházející z *metody bariérových funkcí*, viz např. [3, 1]. Základní idea je taková, že místo minimalizace funkce  $Q_{P_i}$  budeme minimalizovat kritériální funkci  $\Phi_{P_i}$ :

$$\Phi_{P_i}(\mathbf{x}, \alpha) := \alpha Q_{P_i}(\mathbf{x}) + B_{P_i}(\mathbf{x}), \quad \mathbf{x} \in \overset{\circ}{\mathcal{K}}_{P_i}^{ad}, \quad (3.115)$$

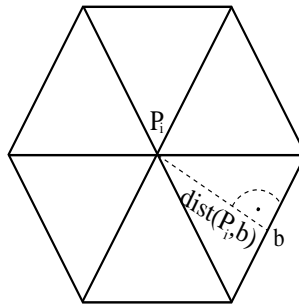
kde  $\mathbf{x}$  je poloha vrcholu  $P_i$ ,  $B_{P_i}$  je bariérová funkce příslušející vrcholu  $P_i$ , parametr  $\alpha > 0$  je váha  $Q_{P_i}$  a  $\overset{\circ}{\mathcal{K}}_{P_i}^{ad}$  značí vnitřek množiny  $\mathcal{K}_{P_i}^{ad}$ . Bariérová funkce  $B_{P_i}$  je kladná spojitá funkce definovaná na  $\mathcal{K}_{P_i}^{ad}$ , od které požadujeme následující:

$$\mathbf{x} \rightarrow \mathbf{y} \in \partial \mathcal{K}_{P_i}^{ad} \Rightarrow B_{P_i}(\mathbf{x}) \rightarrow \infty, \quad \text{kde } \mathbf{x} \in \overset{\circ}{\mathcal{K}}_{P_i}^{ad}. \quad (3.116)$$

Tento požadavek nám při minimalizaci funkce  $\Phi_{P_i}$  zajistí, že uzel  $P_i$  zůstane uvnitř povolené oblasti  $\mathcal{K}_{P_i}^{ad}$ . V naší práci používáme následující bariérovou funkci:

$$B_{P_i}(\mathbf{x}) := \sum_{b \subset \partial \mathcal{K}_{P_i}^{ad}} \frac{1}{\text{dist}(\mathbf{x}, b)}, \quad (3.117)$$

kde  $\text{dist}(\mathbf{x}, b)$  je vzdálenost vrcholu  $P_i$  o souřadnicích  $\mathbf{x}$  od krajní hrany  $b \in \partial \mathcal{K}_{P_i}^{ad}$ , viz obrázek 3.44.



Obrázek 3.44: Určení vzdálenosti hrany  $b$  od vrcholu  $P_i$ .

Pro minimalizaci kritériální funkce (3.115) ve 2D, je použit následující algoritmus, kde na vstupu předpokládáme maximální počet iterací nad jedním uzlem *TSTEP*, maximální počet globálních iterací, tj. iterací nad celou sítí, *MAXITER* a parametry metody BFGS(viz dále):

### Algoritmus 3.2. Algoritmus metody s bariérovou funkcí

1. Spočtíme  $Q = \sum_{P_i \in \sigma_h} Q_{P_i}(P_i)$ , kde  $\sigma_h$  je množina vrcholů sítě.
2. Pro  $\forall P_i \in \sigma_h^{int}$ , kde  $\sigma_h^{int} \subset \sigma_h$  je množina nehraničních vrcholů sítě:
  - (a) Zvolme  $\alpha = B_{P_i}(P_i)/Q_{P_i}(P_i)$ .
  - (b) Přesuňme nehraniční vrchol  $P_i$  na novou polohu  $\tilde{P}_i$ , získanou minimalizací  $\Phi_{P_i}(\mathbf{x}, \alpha)$ , viz (3.115),  $\mathbf{x} \in \mathcal{K}_{P_i}^{ad}$  metodou BFGS.
  - (c) Spočtíme  $\alpha = \alpha\beta$ , kde  $\beta > 1$  (nejčastěji volíme  $\beta = 10$ ).
  - (d) Opakujme znovu od bodu (b), dokud počet iterací nad jedním uzlem není roven předepsanému počtu iterací nad jedním uzlem  $TSTEP$ , nebo pokud nedojde ke zhoršení  $Q_{P_i}$ , tedy  $Q_{\tilde{P}_i}(\tilde{P}_i) \geq Q_{P_i}(P_i)$ .
3. Spočtíme  $\tilde{Q} = \sum_{\tilde{P}_i \in \sigma_h} Q_{\tilde{P}_i}(\tilde{P}_i)$ .
4. Pokud není dosaženo maximálního počtu globálních iterací  $MAXITER$  a zároveň platí  $\tilde{Q} < Q$ , opakujme od bodu 1.

Poznamenejme, že v případě, že dojde k přerušení výpočtu z důvodu zhoršení kvality uzlu, použije se taková poslední pozice, která měla nižší nebo stejnou hodnotu kvality uzlu  $Q_{P_i}$  jako počáteční pozice.

### Metoda BFGS

Pro vlastní minimalizaci byla použita *metoda BFGS*<sup>39</sup> [32], což je kvazinewtonovská metoda, tj. metoda, kde je místo inverze Hessovy<sup>40</sup> matice minimalizované funkce  $f(\mathbf{x})$  použita aproximace pomocí symetrické matice  $\mathbb{S}$ . Nejprve si zavedme značení  $f_k = f(\mathbf{x}_k)$ ,  $\mathbf{d}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ , gradient  $\mathbf{g}_k = (\nabla f(\mathbf{x}_k))^T$ , rozdíl gradientů  $\mathbf{q}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$  a spádový směr  $\mathbf{s}_k = -\mathbb{S}_k \mathbf{g}_k$ . Iterační předpis metody je následující:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{s}_k, \quad k = 0, 1, \dots, \quad (3.118)$$

kde  $\eta_k$  je parametr určující délku kroku a  $\mathbb{S}_k$  je symetrická pozitivně definitní matice, v našem případě volíme jako  $\mathbf{x}_0$  polohu vrcholu  $P_i$ . Výpočet matice  $\mathbb{S}_k$  se provádí iteračním předpisem:

$$\mathbb{S}_{k+1} = \gamma \left( \mathbb{S}_k + \left( \frac{\rho}{\gamma} + \frac{\mathbf{q}_k^T \mathbb{S}_k \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{d}_k} \right) \frac{\mathbf{d}_k \mathbf{d}_k^T}{\mathbf{q}_k^T \mathbf{d}_k} - \frac{\mathbf{d}_k \mathbf{q}_k^T \mathbb{S}_k + \mathbb{S}_k \mathbf{q}_k \mathbf{d}_k^T}{\mathbf{q}_k^T \mathbf{d}_k} \right), \quad (3.119)$$

kde parametr  $\rho$  je stabilizační parametr a parametr  $\gamma$  je škálovací parametr. Jako počáteční matice  $\mathbb{S}_0$  se použije jednotková matice  $\mathbb{I}$ . Stabilizační parametr  $\rho$  určíme [32]:

$$\rho := \begin{cases} \text{pokud } \frac{1}{2} \frac{\mathbf{d}_k^T \mathbf{q}_k}{f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) + \mathbf{d}_k^T \mathbf{g}_{k+1}} \in \langle 0.01, 100 \rangle, \\ \text{jinak } 1. \end{cases} \quad (3.120)$$

<sup>39</sup> Autoři Broyden, Fletcher, Goldfarb a Shannoa

<sup>40</sup> Jde o použití Newtonovy metody na rovnici  $\nabla f(\mathbf{x}) = 0$ , kde  $f(\mathbf{x})$  je minimalizovaná funkce.



Škálovací parametr  $\gamma$  je daný předpisem [32]:

$$\gamma := \begin{cases} \text{pokud } \rho \frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{g}_k^T \mathbb{S}_k \mathbf{g}_k} \in \langle 1, 10 \rangle, \\ \text{jinak } 1. \end{cases} \quad (3.121)$$

Pro určení vhodné délky kroku  $\eta_k$  je použita *metoda zlatého řezu*, kde pro horní mez délky kroku  $\eta_k$  je použita nejmenší hodnota vzdálenosti vrcholu  $P_i$  od krajních hran  $b \in \partial \mathcal{K}_{P_i}$ . Poznamenejme, že v celém textu předpokládáme, že metoda je implementována tím způsobem, že nedovolí vzrůst hodnoty minimalizované funkce, v našem případě  $\Phi_{P_i}(\mathbf{x}, \alpha)$ , nad její počáteční hodnotu.

Předpokládejme, že na vstupu algoritmu je zadán maximální počet iterací metody BFGS *MAXITBFGS*, maximální počet iterací metody zlatého řezu, pro určení  $\eta_k$ , *MAXITG* a minimální hodnota normy gradientu  $\varepsilon_{BFGS}$  pro zastavení. Vlastní algoritmus metody BFGS je následující.

### Algoritmus 3.3. Algoritmus metody BFGS

1. Zvolme jako  $\mathbf{x}_0$  polohu vrcholu  $P_i$ , spočtěme gradient  $\mathbf{g}_0$ , matici  $\mathbb{S}_0 = \mathbb{I}$ .
2. Pokud  $\|\mathbf{g}_k\| \geq \varepsilon_{BFGS}$  a zároveň není dosaženo maximálního počtu iterací *MAXITBFGS*, pokračujeme dále.
3. Spočtěme spádový směr  $\mathbf{s}_k = -\mathbb{S}_k \mathbf{g}_k$ , dále nalezneme optimální délku kroku  $\eta_k$  a spočtěme pomocí (3.118) novou pozici  $\mathbf{x}_{k+1}$ .
4. Spočtěme hodnoty  $f_{k+1}$ ,  $\mathbf{g}_{k+1}$ ,  $\mathbf{d}_k$  a  $\mathbf{q}_k$ . Dále ze vztahů (3.120) a (3.121), určíme stabilizační parametr  $\rho$  a škálovací parametr  $\gamma$  a pomocí (3.119) spočtěme novou matici  $\mathbb{S}_{k+1}$  a pokračujeme bodem 2.

### 3.3.2 Metoda s kontrolou úhlů

Algoritmus předchozí metody je značně komplikovaný a obsahuje řadu optimalizačních cyklů<sup>41</sup>, navíc předchozí metoda neumožňuje kontrolu předepsaného minimálního úhlu  $\delta_{min} > 0$ , jež mají svírat hrany každého z objemů viz (3.111). Z tohoto důvodu jsme navrhli vlastní metodu využívající pro udržení nehraničního vrcholu  $P_i$  v přípustné oblasti  $\mathcal{K}_{P_i}^{ad}$  kontrolu úhlů.

Zavedme si funkci  $\theta(D_j)$ ,  $D_j \in \mathcal{T}_h$ , která přiřadí objemu  $D_j$  nejmenší úhel, který svírají hrany tohoto objemu. Uvažujme, že než uzel při pohybu vyjde z povolené oblasti  $\mathcal{K}_{P_i}^{ad}$ , musí nastat ta situace, že  $\theta(D_j) < \delta_{min}$ ,  $D_j \in \mathcal{K}_{P_i}$ , tj. hrany některého z objemů  $D_j \in \mathcal{K}_{P_i}$  spolu svírají menší úhel než  $\delta_{min}$ . Z tohoto důvodu použijeme následující kriteriální funkci:

$$\Phi_{P_i}(\mathbf{x}, \delta, \delta_{min}) := \frac{Q_{P_i}(\mathbf{x})}{\Psi_{P_i}(\delta, \delta_{min})}, \quad (3.122)$$

kde  $\Psi_{P_i}$  je funkce, kterou používáme pro kontrolu úhlu  $\delta$ , tj. nejmenšího úhlu v objemech tvořících  $\mathcal{K}_{P_i}$ :

$$\delta = \min_{D_j \in \mathcal{K}_{P_i}} \theta(D_j), \quad (3.123)$$

<sup>41</sup>Pro každý uzel se provádí algoritmus metody bariérových funkcí, jehož jádrem je cyklus metody BFGS. V metodě BFGS je použita jednorozměrná minimalizace pomocí metody zlatého řezu.

tak, aby byl větší než jistá předepsaná hodnota  $\delta_{min}$ . Funkce  $\Psi_{P_i}$  je diferencovatelná a klademe na ni následující požadavky:

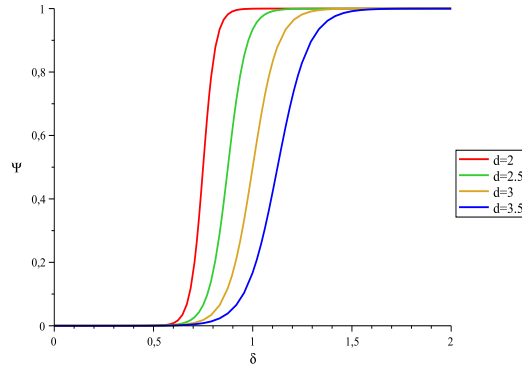
$$\begin{aligned}\Psi_{P_i}(\delta, \delta_{min}) &\longrightarrow 0 \text{ když } \delta \longrightarrow \delta_{min}, \\ \Psi_{P_i}(\delta, \delta_{min}) &\longrightarrow 1 \text{ když } \delta \longrightarrow d\delta_{min} \text{ kde } d > 1.\end{aligned}\tag{3.124}$$

Tyto vlastnosti funkce  $\Psi_{P_i}$  vedou k tomu, že pokud se při pohybu vrcholu  $P_i$  v nějakém směru začne nejmenší úhel  $\delta > \delta_{min}$  blížit k hodnotě  $\delta_{min}$ , tj.  $\delta < d\delta_{min}$ , pak  $\Psi_{P_i}$  klesá, čili dochází k nárůstu minimalizované kriteriální funkce  $\Phi_{P_i}$  (3.122) a tento směr pohybu vrcholu je při minimalizaci brán jako nevýhodný a nepoužije se.

Příkladem takové funkce může být hyperbolický tangets<sup>42</sup>, kde argument je upraven tak, že pokud  $\delta = \delta_{min}$ , má argument hodnotu  $-a$ , pokud  $\delta = d\delta_{min}$ ,  $d > 1$ , tak argument má hodnotu  $a$ , předpokládáme  $a > 0$ . Aby byly splněny požadavky (3.124) použijeme funkci  $(\tanh(x) + 1)/2$  a dostáváme:

$$\Psi_{P_i}(\delta, \delta_{min}) := \frac{\tanh\left(\frac{2a}{(d-1)}\left(\frac{\delta}{\delta_{min}} - \frac{(d+1)}{2}\right)\right) + 1}{2},\tag{3.125}$$

parametr  $a > 0$  ovlivňuje hodnotu  $\Psi_{P_i}(\delta_{min}, \delta_{min})$ , parametr  $d$  ovlivňuje rychlost přechodu  $\Psi_{P_i}$  mezi 0 a 1, máme tedy možnost ovlivnit, kdy už začíná být úhel  $\delta$  nepřijatelný, viz obrázek 3.45.



Obrázek 3.45: Průběh funkce  $\Psi_{P_i}$  pro  $\delta_{min} = 0.5$ ,  $a = 4$ . Zde je vidět, jak se vzrůstajícím  $d$  se rozšiřuje přechod mezi 0 a 1.

Pro vlastní minimalizaci parametru kvality všech nehraničních uzlů byl použit následující algoritmus, kde na vstupu předpokládáme maximální počet globálních iterací *MAXITER* a parametry metody BFGS:

#### Algoritmus 3.4. Algoritmus metody s kontrolou úhlů

1. Spočtíme  $Q = \sum_{P_i \in \sigma_h} Q_{P_i}(P_i)$ , kde  $\sigma_h$  je množina vrcholů sítě.
2. Pro  $\forall P_i \in \sigma_h^{int}$ , kde  $\sigma_h^{int} \subset \sigma_h$  je množina nehraničních vrcholů sítě:
  - Přesuňme nehraniční vrchol  $P_i$  na novou polohu  $\tilde{P}_i$ , získanou minimalizací  $\Phi_{P_i}(\mathbf{x}, \delta, \delta_{min})$ , viz (3.122),  $\mathbf{x} \in \mathcal{K}_{P_i}^{ad}$ , metodou BFGS.

<sup>42</sup>Motivací nám byly aktivační funkce užívané v neuronových sítích, viz např. [34].

3. Spočtáme  $\tilde{Q} = \sum_{\tilde{P}_i \in \sigma_h} Q_{\tilde{P}_i}(\tilde{P}_i)$ .
4. Pokud není dosaženo maximálního počtu globálních iterací *MAXITER* a zároveň platí  $\tilde{Q} < Q$ , opakujeme od bodu 1.

Poznamenejme, že v případě, že dojde k přerušení výpočtu z důvodu zhoršení  $\Phi_{P_i}$ , použije se taková poslední pozice, která měla nižší nebo stejnou hodnotu  $\Phi_{P_i}$  jako počáteční pozice. Pro nalezení nové polohy uzlu  $P_i$  je stejně, jako u předchozí metody, použita metoda BFGS s jednorozměrnou minimalizací pomocí metody zlatého řezu.

### 3.3.3 Metoda s opravným cyklem

Výše použité postupy minimalizace parametru kvality vrcholu  $Q_{P_i}$  jsou založeny na robustních numerických metodách optimalizace s omezením a nevyužívají dalších specifických vlastností úlohy, jako je například to, že  $\partial\mathcal{K}_{P_i}$  je tvořena množinou úseček. Dále při numerických experimentech bylo zjištěno, že, pokud zůstává aproximace Hessovy matice  $\mathbb{M}$  (3.106) během minimalizace konstantní, stačí provést nad každým vrcholem  $P_i$  pomocí metody BFGS pouze jednu nebo dvě iterace. Z tohoto důvodu použijme alternativního postupu založeného na metodě největšího spádu, který má nižší řád konvergence, ale je výpočetně jednodušší z toho důvodu, že není třeba konstruovat matici  $\mathbb{S}_k$ , viz (3.119), což vůči předchozím metodám zmenší dobu nutnou pro běh adaptace. Dalším důvodem je poměrně snadná implementace pohybu vrcholů sítě po hranici.

V iteračním schématu metody BFGS (3.118) budeme uvažovat, že spádový směr  $\mathbf{s}_k$  odpovídá opačnému směru ke gradientu  $\mathbf{g}_k$ , tedy iterační matice  $\mathbb{S}_k$  je v průběhu celého výpočtu<sup>43</sup> rovna jednotkové matici  $\mathbb{I}$ . Udržení polohy pohybujícího se vrcholu  $P_i$  v oblasti  $\mathcal{K}_{P_i}^{ad}$  docílíme vhodnou volbou horní meze délky kroku  $\eta_k$ , viz obrázek 3.46.

Pro nehraniční uzly je tato horní mez  $\eta_k^{max}$  určena následovně:

#### Algoritmus 3.5. Algoritmus nalezení $\eta^{max}$

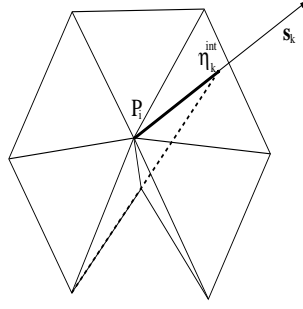
1. Nalezneme nejmenší kladnou hodnotu parametru  $\eta$ , pro kterou má vektor  $\mathbf{x}_k + \eta\mathbf{s}_k$  průsečík s hranou  $e \in \partial\mathcal{K}_{P_i}$ , respektive s jejím prodloužením, tj. přímkou jejíž restrikce je daná hrana, tuto hodnotu označme  $\eta_k^{int}$ .
2. Spočtáme  $\eta_k^{max}$ :

$$\eta_k^{max} := \gamma_r \eta_k^{int}, \quad \gamma_r \in (0, 1). \quad (3.126)$$

Tento způsob nám ovšem nezaručí to, že v žádném z objemů tvořících  $\mathcal{K}_{P_i}$  nebude menší úhel, než nejmenší předepsaný úhel  $\delta_{min}$ <sup>44</sup>. Po posunu vrcholu je proto nutné určit nejmenší úhel  $\delta$ , viz (3.123), a v případě, že je menší než  $\delta_{min}$  snížíme hodnotu  $\eta_k^{max}$  a celý postup opakujeme tak dlouho, než je splněna podmínka  $\delta \geq \delta_{min}$ , nebo je dosaženo maximálního počtu opravných iterací. Protože v tomto cyklu opravujeme již určenou novou polohu vrcholu  $P_i$ , nazýváme naši

<sup>43</sup>Při implementaci, samozřejmě neprovádíme násobení jednotkovou maticí.

<sup>44</sup>Toto by šlo vzít v potaz a určit hodnotu  $\eta_k^{int}$  tak, aby byla splněna i tato podmínka. Vedlo by to ovšem k tomu, že bychom byli nuceni řešit soustavu nelineárních rovnic v každém iteračním kroku a tím ke zpomalení výpočtu.



Obrázek 3.46: Pokud se při pohybu vrcholu  $P_i$  ve směru  $\mathbf{s}_k$  bude vrchol držet v tučně naznačené oblasti, tj.  $\eta \in \langle 0, \eta_k^{int} \rangle$ , je zaručeno, že vrchol po přesunu zůstane v oblasti  $\mathcal{K}_{P_i}^{ad}$ .

metodu metodou s opravným cyklem. Je nutné si uvědomit, že díky redukci pomocí  $\gamma_r$  v (3.126) a snižování  $\eta_k^{max}$  v opravném cyklu, může dojít k vynechání minima v daném směru  $\mathbf{s}_k$ , tedy nalezneme pouze suboptimální řešení<sup>45</sup>.

Předpokládejme, že je zadán maximální počet iterací  $MAXITGRAD$ , maximální počet opravných iterací  $MAXITREP$ , minimální hodnota normy gradientu  $\varepsilon_{GRAD}$  pro zastavení a redukční koeficient  $\gamma_r$ . Algoritmus pro pohyb nehraničního vrcholu je následující:

### Algoritmus 3.6. Pohyb nehraničního vrcholu s opravným cyklem

1. Vezměme nehraniční vrchol  $P_i \in \sigma_h^{int}$ , kde  $\sigma_h^{int} \subset \sigma_h$  je množina nehraničních vrcholů sítě, a zvolme jako  $\mathbf{x}_0$  polohu vrcholu  $P_i$ .
2. Spočtíme gradient  $\mathbf{g}_k$  funkce  $Q_{P_i}(\mathbf{x}_k)$ , zvolme spádový směr  $\mathbf{s}_k = -\mathbf{g}_k$ .
3. Pokud  $\|\mathbf{g}_k\| \geq \varepsilon_{GRAD}$ , určíme hodnoty  $\eta_k^{int}$  a  $\eta_k^{max}$ , viz (3.126). V opačném případě skončíme.
  - (a) Pomocí metody zlatého řezu nalezneme minimum funkce  $Q_{P_i}(\mathbf{x}_k + \eta_k \mathbf{s}_k)$  ve směru  $\mathbf{s}_k$ , kde jako horní mez pro parametr  $\eta_k$  slouží hodnota  $\eta_k^{max}$ , tento bod označíme jako  $\mathbf{x}_{new}$ .
  - (b) V případě, že  $\delta < \delta_{min}$  a není dosaženo maximálního počtu opravných iterací  $MAXITREP$ , snižme  $\eta_k^{max} = \gamma_r \eta_k^{max}$  a pokračujme bodem (a). V případě, že je dosaženo maximálního počtu opravných iterací ukončíme algoritmus.
4. Položíme  $\mathbf{x}_{k+1} = \mathbf{x}_{new}$  a, jestliže není dosaženo maximálního počtu iterací nad jedním vrcholem  $MAXITGRAD$ , pokračujme bodem 2.

Pro pohyb uzlů, které leží na hranici oblasti  $\partial\Omega$ , použijeme obdobný algoritmus. Předpokládejme, že pro zvolené dvojice bodů  $a_j, b_j \in \mathbb{R}$ , kde  $a_j < b_j$  lze hranici  $\partial\Omega$  rozdělit na segmenty  $\Gamma_j \in \partial\Omega$ ,  $j = 1, 2, \dots$  takové, že lze psát:

$$\begin{aligned} \Gamma_j &= \{(x_1, \phi_j(x_1)) \in \mathbb{R}^2; x_1 \in \langle a_j, b_j \rangle\}, \text{ resp.} & (3.127) \\ \Gamma_j &= \{(\phi_j(x_2), x_2) \in \mathbb{R}^2; x_2 \in \langle a_j, b_j \rangle\}, \end{aligned}$$

<sup>45</sup>Efektivita tohoto postupu závisí na způsobu zmenšování  $\eta_k^{max}$  a volbě redukčního koeficientu  $\gamma_r$ .



- (a) Pomocí modifikované metody zlatého řezu určíme nové polohy uzlů  $\mathbf{x}_{new,L}$ ,  $\mathbf{x}_{new,R}$  a zvolíme  $\mathbf{x}_{new} = \arg \min(Q_{P_i}(\mathbf{x}_{new,L}), Q_{P_i}(\mathbf{x}_{new,R}))$ .
- (b) V případě, že  $\delta < \delta_{min}$ , snižme příslušné  $\eta_k^{max}$  pomocí:  $\eta_k^{max} = \gamma_r \eta_k^{max}$  a v případě, že není dosaženo maximálního počtu opravných iterací *MAXITREP* pokračujeme bodem (a). V případě, že je dosaženo maximálního počtu opravných iterací ukončíme algoritmus.

4. Položíme  $\mathbf{x}_{k+1} = \mathbf{x}_{new}$  a, jestliže není dosaženo maximálního počtu iterací nad jedním vrcholem *MAXITGRAD*, pokračujeme bodem 2.

Pro vlastní minimalizaci parametru kvality všech uzlů byl použit následující algoritmus, kde na vstupu předpokládáme maximální počet globálních iterací *MAXITER* a parametry výše zmíněných algoritmů 3.6 a 3.7, počet opakování nad jedním vrcholem *MAXITGRAD*, maximální počet opravných iterací *MAXITREP* a redukční koeficient  $\gamma_r$ :

### Algoritmus 3.8. Algoritmus metody s opravným cyklem

1. Spočítáme  $Q = \sum_{P_i \in \sigma_h} Q_{P_i}(P_i)$ , kde  $\sigma_h$  je množina všech vrcholů sítě.
2. Pro  $\forall P_i \in \sigma_h$ :
  - (a) Přesuňme  $P_i$  na novou polohu  $\tilde{P}_i$ , získanou algoritmem 3.7 pro hraniční vrcholy a algoritmem 3.6 pro vrcholy nehraniční.
3. Spočítáme  $\tilde{Q} = \sum_{\tilde{P}_i \in \sigma_h} Q_{\tilde{P}_i}(\tilde{P}_i)$ .
4. Pokud není dosaženo maximálního počtu globálních iterací *MAXITER* a zároveň platí  $\tilde{Q} < Q$ , opakujeme od bodu 1.

### 3.3.4 Minimalizace parametru kvality v 1D

Zde si popíšeme naši původní metodu pro pohyb uzlů v 1D, založenou na anisotropní adaptivitě. Obdobně jako v kapitole (3.1.4) předpokládejme, že máme 1D síť konečných objemů  $\mathcal{D}_h = \{D_i\}_{i=1,\dots,J}$  pokrývající interval  $[a, b]$ , kde  $a < b$ ,  $J$  je počet objemů. Nechť uzly  $x_i$  a  $x_{i+1}$ , kde  $x_i < x_{i+1}$ , tvoří objem  $D_i$ .

Připomeňme si, že jako  $\mathbf{w}_h^*$  značíme lineární rekonstrukci<sup>46</sup> získanou z po částech konstantního numerického řešení  $\mathbf{w}_h^k$  Eulerových rovnic (1.17) v čase  $t_k$ , pomocí vztahu (3.82). Zkonstruujeme z  $\mathbf{w}_h^*$  po částech lineární fyzikální veličinu  $u_h^* \in \mathcal{L}_{\mathcal{D}_h}$ , kde  $\mathcal{L}_{\mathcal{D}_h}$  představuje prostor po částech lineárních funkcí na  $\mathcal{D}_h$ . Poznamenejme, že v našich numerických experimentech budeme používat přímo jednu ze složek  $\mathbf{w}_h^*$ . Tato veličina  $u_h^* \in \mathcal{L}_{\mathcal{D}_h}$  představuje aproximaci veličiny  $u \in W$  zkonstruované z přesného řešení  $\mathbf{w}(\cdot, t_k)$  Eulerových rovnic (1.17) v čase  $t_k$ , kde  $W$  je vhodný prostor funkcí.

Nyní lze, obdobně jako v kapitole 3.2.1, sestavit aproximaci druhé derivace  $u_h^{*xx}(x_i)$ :

$$\left( \frac{d^2 u}{dx^2}, \varphi \right) \approx (u_h^{*xx}, \varphi) = \left[ \frac{du_h^*}{dx} \varphi \right]_a^b - \left( \frac{du_h^*}{dx}, \frac{d\varphi}{dx} \right) \quad \forall \varphi \in \mathcal{L}_{\mathcal{D}_h}, \quad (3.128)$$

<sup>46</sup>Pro jednoduchost budeme horní index  $k$  vynechávat.

kde  $(\cdot, \cdot)$  rozumíme skalární součin na  $L^2(\Omega)$  a  $\varphi$  je testovací funkce. Použijme následující množinu funkcí:

$$\{\varphi_i; \varphi_i \in \mathcal{L}_{\mathcal{D}_h}, \varphi_i(x_j) = \delta_{ij}, i, j = 1, \dots, J+1\}, \quad (3.129)$$

kde  $x_j$  je některý z uzlů sítě,  $\delta_{ij}$  je Kroneckerovo delta. Analogickými úpravami jako ve výše zmíněné kapitole, dostáváme pro nehraniční uzly vztah pro výpočet aproximace druhé derivace<sup>47</sup>:

$$u_h^{*xx}(x_i) = \frac{2}{x_{i+1} - x_{i-1}} \left( \frac{u_{i+1}^* - u_i^*}{x_{i+1} - x_i} - \frac{u_i^* - u_{i-1}^*}{x_i - x_{i-1}} \right), \quad i = 2, \dots, J, \quad (3.130)$$

kde pomocí  $u_i^*$  je označena hodnota  $u_h^*(x_i)$ . Hodnoty  $u^{*xx}$  v krajních uzlech  $x_1$ , resp.  $x_{J+1}$  získáme lineární extrapolací z hodnot  $u^{*xx}(x_2)$  a  $u^{*xx}(x_3)$ , resp. z hodnot  $u^{*xx}(x_{J-1})$  a  $u^{*xx}(x_J)$ . Modifikaci Hessovy matice  $\mathbb{M}$  v  $i$ -tém uzlu sítě, vztah (3.106), lze v 1D psát:

$$M_i := \bar{c} \left[ 1 + \frac{\varepsilon_1 |u^{*xx}(x_i)|}{\varepsilon_2 + |u^{*xx}(x_i)|} \right], \quad (3.131)$$

kde význam parametrů  $\varepsilon_1$  a  $\varepsilon_2$  je stejný jako v kapitole 3.2.1, hodnota  $\bar{c}$  je dána vztahem (3.109). Při praktické realizaci budeme používat zhlazení této hodnoty:

$$\tilde{M}_i = \frac{1}{4} (M_{i-1} + 2M_i + M_{i+1}), \quad (3.132)$$

kde pro krajní uzly  $x_i$  používáme přímo hodnotu  $M_i$ . Jde o analogii zhlazování monitorovacích funkcí  $\mathbb{G}$  u variačních metod, viz vztah (3.55).

Zavedme si funkci  $Q$ , jako součet kvalit jednotlivých uzlů:

$$Q(\mathbf{x}) := \sum_{i=1}^{J+1} Q_{x_i}(x_i), \quad (3.133)$$

kde  $\mathbf{x} = (x_1, \dots, x_{J+1})^\top$ , což po dosazení z definice kvality uzlu  $Q_{x_i}$ , viz vztah (3.114), vede na:

$$Q(\mathbf{x}) = 2 \sum_{i=2}^{J+1} \left( \sqrt{\frac{M_i + M_{i-1}}{2}} (x_i - x_{i-1}) - c_1 \right)^2, \quad (3.134)$$

kde hodnota  $c_1$  je dána vztahem (3.96). Minimalizujme tuto funkci za podmínek  $x_1 = a$  a  $x_{J+1} = b$ . Naším cílem je získat soustavu lineárních rovnic pro pohyb uzlů. Spočteme parciální derivace podle volných proměnných, kde uvažujeme  $M_i$  konstantní<sup>48</sup>, dostáváme:

$$\begin{aligned} \frac{\partial Q}{\partial x_i}(\mathbf{x}) &= 4((M_{i,i-1}(x_i - x_{i-1}) - c_1) M_{i,i-1} \\ &\quad - (M_{i+1,i}(x_{i+1} - x_i) - c_1) M_{i+1,i}), \quad i = 2, \dots, J, \end{aligned} \quad (3.135)$$

<sup>47</sup>Poznamenejme, že obdobný vztah bychom získali při použití centrálních diferencí na funkci  $u$ .

<sup>48</sup>Ve skutečnosti  $M_i = M_i(u^{*xx}(x_i))$ , pokud bychom pokračovali naznačeným postupem, tak bychom získali soustavu nelineárních rovnic, jejíž řešení by bylo z hlediska času náročnější.

kde  $M_{i,i-1} = \sqrt{\frac{M_i + M_{i-1}}{2}}$ . Pro nalezení stacionárního bodu položíme tyto derivace rovny nule a použijeme výše zmíněné podmínky<sup>49</sup>. Po úpravách získáme následující soustavu rovnic pro určení poloh vnitřních uzlů:

$$\begin{pmatrix} \beta_2 & \alpha_2 & 0 & 0 & \dots & 0 \\ \alpha_2 & \beta_3 & \alpha_3 & 0 & \dots & 0 \\ 0 & \alpha_3 & \beta_4 & \alpha_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_{J-2} & \beta_{J-1} & \alpha_{J-1} \\ 0 & 0 & \dots & 0 & \alpha_{J-1} & \beta_J \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{J-1} \\ x_J \end{pmatrix} = \begin{pmatrix} b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_{J-1} \\ b_J \end{pmatrix}, \quad (3.136)$$

kde platí, že

$$\beta_i = M_{i-1} + 2M_i + M_{i+1} \Rightarrow \beta_i > 0, \quad i = 2, \dots, J, \quad (3.137)$$

a

$$\alpha_i = -(M_i + M_{i+1}) \Rightarrow \alpha_i < 0, \quad i = 2, \dots, J. \quad (3.138)$$

Složky vektoru pravých stran jsou následující:

$$\begin{aligned} b_2 &= 2c_1 \left( \sqrt{\frac{-\alpha_1}{2}} - \sqrt{\frac{-\alpha_2}{2}} \right) - \alpha_1 a, \\ b_i &= 2c_1 \left( \sqrt{\frac{-\alpha_{i-1}}{2}} - \sqrt{\frac{-\alpha_i}{2}} \right), \quad i = 3, \dots, J-1, \\ b_J &= 2c_1 \left( \sqrt{\frac{-\alpha_{J-1}}{2}} - \sqrt{\frac{-\alpha_J}{2}} \right) - \alpha_J b. \end{aligned} \quad (3.139)$$

Dalším derivováním vztahů (3.135) obdržíme Hessovu matici funkce  $Q(\mathbf{x})$  která, až na vynásobení kladnou konstantou, odpovídá matici soustavy (3.136).

Vztahy (3.137) a (3.138) určují následující vazby mezi prvky  $\alpha_i$  a  $\beta_i$  matice:

$$|\beta_2| > |\alpha_2|, \quad |\beta_J| > |\alpha_{J-1}| \quad (3.140)$$

$$\alpha_{i-1} + \beta_i + \alpha_i = 0, \quad i = 3, \dots, J-1. \quad (3.141)$$

Díky výše uvedeným vazbám  $\alpha_i$  a  $\beta_i$ , lze matici v (3.136) psát jako:

$$\mathbb{A} = \begin{pmatrix} \beta_2 & \alpha_2 & 0 & 0 & \dots & 0 \\ \alpha_2 & -(\alpha_2 + \alpha_3) & \alpha_3 & 0 & \dots & 0 \\ 0 & \alpha_3 & -(\alpha_3 + \alpha_4) & \alpha_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_{J-2} & -(\alpha_{J-2} + \alpha_{J-1}) & \alpha_{J-1} \\ 0 & 0 & \dots & 0 & \alpha_{J-1} & \beta_J \end{pmatrix}. \quad (3.142)$$

V dalším uvažujeme převod této matice pomocí Gaussovy eliminace na horní trojúhelníkový tvar. Je možné ukázat<sup>50</sup>, že, pokud lze nějaký diagonální prvek  $\tilde{\beta}_i$  výše

<sup>49</sup>Tento postup je ekvivalentní použití metody Lagrangeových multiplikátorů.

<sup>50</sup>Stačí pouze dosadit do matice (3.142) a použít Gaussovu eliminaci.



uvedené matice zapsat ve tvaru  $\tilde{\beta}_i = -(\alpha_i + \delta_i)$ , kde  $\alpha_i < 0$ ,  $\delta_i < 0$ , tak i následující diagonální prvek  $\tilde{\beta}_{i+1}$  lze psát ve tvaru  $\tilde{\beta}_{i+1} = -(\alpha_{i+1} + \delta_{i+1})$ , kde  $\alpha_{i+1} < 0$ ,  $\delta_{i+1} < 0$ . Z podmínky (3.140) a (3.137) pro  $\beta_2$  a (3.138) pro  $\alpha_2$  je zřejmé, že  $\beta_2$  takto zapsat lze, tedy indukcí dostáváme, že matice (3.142) po převodu na horní trojúhelníkový tvar má na diagonále pouze kladné prvky. Matice je regulární, tedy  $\lambda_i \neq 0$ ,  $i = 1, \dots, J - 1$ .

Protože matice (3.142) je symetrická, diagonálně dominantní, s kladnými prvky na diagonále, tak z Gershgorinovy věty pro její vlastní čísla plyne  $\lambda_i \geq 0$ ,  $i = 1, \dots, J - 1$ . Toto, spolu z předcházejícím, dává její pozitivní definitnost. Máme tedy zaručeno, že funkce  $Q(\mathbf{x})$  má v bodě nalezeném řešení soustavy (3.136) minimum<sup>51</sup>.

Pro řešení soustavy (3.136) budeme používat *Gaussovu-Seidelovu* a *Jacobiovu* iterační metodu. Díky tomu, že matice soustavy je symetrická a pozitivně definitní, tak u Gaussovu-Seidelovy metody je zaručena konvergence [36, Theorem 4.5]. Protože platí, že v případě třídiagonálních matic zmíněné metody, buď obě konvergují, nebo obě nekonvergují [36, str.131, vztah 4.18], tak konverguje i Jacobiova metoda.

Abychom zaručili, že v každé iteraci  $\nu$  bude splněna podmínka:

$$a = x_1^\nu < x_2^\nu < \dots < x_{J+1}^\nu = b, \nu = 1, 2, \dots, \quad (3.143)$$

budeme používat následujícího postupu. Nejprve spočteme  $\tilde{x}_i^{\nu+1}$  pomocí Jacobiovy metody:

$$\begin{aligned} \tilde{x}_2^{\nu+1} &= \frac{1}{\beta_2} (b_2 - \alpha_3 x_3^\nu), \\ \tilde{x}_i^{\nu+1} &= \frac{1}{\beta_i} (b_i - \alpha_{i-1} x_{i-1}^\nu - \alpha_{i+1} x_{i+1}^\nu), \quad i = 3, \dots, J - 1, \\ \tilde{x}_J^{\nu+1} &= \frac{1}{\beta_J} (b_J - \alpha_{J-1} x_{J-1}^\nu), \end{aligned} \quad (3.144)$$

resp. Gaussovy-Seidelovy metody pro řešení soustavy (3.136):

$$\begin{aligned} \tilde{x}_2^{\nu+1} &= \frac{1}{\beta_2} (b_2 - \alpha_3 x_3^{\nu+1}), \\ \tilde{x}_i^{\nu+1} &= \frac{1}{\beta_i} (b_i - \alpha_{i-1} x_{i-1}^{\nu+1} - \alpha_{i+1} x_{i+1}^{\nu+1}), \quad i = 3, \dots, J - 1, \\ \tilde{x}_J^{\nu+1} &= \frac{1}{\beta_J} (b_J - \alpha_{J-1} x_{J-1}^{\nu+1}). \end{aligned} \quad (3.145)$$

Polohu  $i$ -tého uzlu v kroku  $\nu + 1$  pak určíme:

$$x_i^{\nu+1} = \begin{cases} \text{pokud } \tilde{x}_i^{\nu+1} \leq x_{i-1}^{\nu+1}, \text{ pak } x_i^{\nu+1} = x_i^\nu - \gamma_r (x_i^\nu - x_{i-1}^{\nu+1}), \\ \text{pokud } \tilde{x}_i^{\nu+1} \in (x_{i-1}^{\nu+1}, x_{i+1}^{\nu+1}), \text{ pak } x_i^{\nu+1} = \tilde{x}_i^{\nu+1}, \\ \text{pokud } \tilde{x}_i^{\nu+1} \geq x_{i+1}^{\nu+1}, \text{ pak } x_i^{\nu+1} = x_i^\nu + \gamma_r (x_{i+1}^{\nu+1} - x_i^\nu), \end{cases} \quad (3.146)$$

kde  $\gamma_r \in (0, 1)$  je redukční koeficient, volíme  $\gamma_r$  blízký jedné. Tato úprava zabrání případnému přesmýknutí uzlů sítě, ale právě omezení pohybu uzlů může zpomalit konvergenci algoritmu.

<sup>51</sup>Za výše uvedeného předpokladu, že  $M_i$  jsou konstantní.

Pro vlastní minimalizaci parametru kvality všech uzlů byl použit následující algoritmus, kde na vstupu předpokládáme maximální počet globálních iterací *MAXITER* a parametry  $\varepsilon_1$  a  $p$ , pro konstrukci aproximací Hessovy matice  $M_i$ , a redukční koeficient  $\gamma_r$ . Jako  $x_i^0$ ,  $i = 1, \dots, J + 1$  použijeme výchozí polohy uzlů sítě.

### Algoritmus 3.9. Algoritmus pohybu uzlů v 1D

1. Zkonstruujeme pomocí (3.82)  $\mathbf{w}_h^*$ , resp.  $u_h^*$  a pak pomocí vztahů (3.130) a (3.131)) spočítáme  $u_h^{*xx}(x_i)$  a  $M_i$  pro všechny uzly sítě.
2. Spočítáme  $Q^\nu = \sum_{i=1}^{J+1} Q_{x_i^\nu}(x_i^\nu)$
3. Pro každý uzel sítě provedeme Jacobiovu (3.144), resp. Gaussovu-Seidelovu iteraci (3.145), která je následována korekcí (3.146).
4. Spočítáme  $Q^{\nu+1} = \sum_{i=1}^{J+1} Q_{x_i^{\nu+1}}(x_i^{\nu+1})$
5. Pokud  $Q^{\nu+1} < Q^\nu$ , opakujeme od bodu<sup>52</sup> 2., dokud není dosaženo maximálního počtu iterací *MAXITER*.

Poznamenejme, že v případě, že není splněna podmínka  $Q^{\nu+1} < Q^\nu$  a dojde k přerušení výpočtu, je použita síť získaná v  $\nu$ -té iteraci.

### 3.3.5 Numerické experimenty v 1D

V této části práce se zaměříme na otestování námi navrženého algoritmu pro adaptaci 1D výpočtové sítě, založeného na minimalizaci parametru kvality vrcholu.

Budeme postupovat obdobně jako v kapitole 3.1.6 při testu variačních metod v 1D. Místo toho, abychom konstruovali po částech lineární fyzikální veličinu  $u_h^* \in \mathcal{L}_{\mathcal{D}_h}$  z numerického řešení Eulerových rovnic (1.17) v 1D, tak si tuto veličinu přímo předepíšeme následujícím způsobem. Použijeme funkci  $u_A$ , viz (3.72), která obsahuje nespojitost, resp.  $u_B$ , viz (3.73), zde se jedná o úlohu, kde je oblast s téměř konstantní hodnotou funkce  $u_B$ . Pomocí vztahu<sup>53</sup> (3.74) zkonstruujeme její po částech konstantní aproximaci a z ní, pomocí vztahu<sup>54</sup> (3.82), po částech lineární fyzikální veličinu  $u_h^* \in \mathcal{L}_{\mathcal{D}_h}$ , kterou používáme pro konstrukci aproximací Hessových matic. Jako výpočetní oblast volíme interval  $[-1, 1]$ .

Sledování konvergence algoritmu budeme určovat z hodnot funkce  $Q$  (3.134) v jednotlivých iteracích  $\nu$  metody. Protože budeme sledovat vliv parametrů  $\varepsilon_1$  a  $p$ , použitých v konstrukci aproximace Hessovy matice  $M_i$ , viz (3.131), na výslednou síť a hodnota  $Q$  vychází pro každou konfiguraci jinak, budeme sledovat  $Q_{norm} =$

<sup>52</sup>Je možné opakovat i od bodu 1., ale vede to k nutnosti v každé iteraci metody znovu spočítat všechny hodnoty z bodu 1., což je značně časově náročné. Tento postup byl použit v kombinaci s transformací numerického řešení pomocí perturbační metody, viz metody *JACVQPMM – PM* a *GSVQPMM – PM* v tabulce 4.1

<sup>53</sup>Pro výpočet integrálu byla použita rutina implementačního jazyka Octave *quad*, což vede k volání metody založené na Gaussově kvadratuře.

<sup>54</sup>Místo stavového vektoru  $\mathbf{w}_i^k$  v (3.82) použijeme přímo hodnot  $u_{i+\frac{1}{2}}$  získané pomocí (3.74).

$Q/Q_{init}$ , kde  $Q_{init}$  je počáteční hodnota  $Q$  na neadaptované síti. Kvalitu adaptace budeme sledovat pomocí následujícího ukazatele chyby:

$$L2E = \frac{1}{J} \sum_{i=1}^J \int_{x_i}^{x_{i+1}} \left( u(x) - u'_{i+\frac{1}{2}} \right)^2 dx, \quad (3.147)$$

kde  $u'_{i+\frac{1}{2}}$  představuje po částech konstantní aproximaci veličiny  $u$  na objemu  $D_i$  v  $\nu$ -té iteraci. Integrál na pravé straně (3.147) je určen pro funkci  $u_A(x)$  analyticky a pro funkci  $u_B(x)$  pomocí numerické kvadratury<sup>55</sup>. Použitý postup je stejný jako v kapitole 3.1.6. Počet objemů byl zvolen  $J = 21$ , počáteční síť obsahovala všechny objemy stejné velikosti. V obou experimentech byl nastaven maximální počet iterací  $MAXITER = 30$ . Předepsaná tolerance  $TOL$  pro výpočet konstanty  $c_1$ , viz (3.96), je zvolena  $TOL = \frac{1}{2}$  a redukční koeficient  $\gamma_r = 0.99$ .

Celý adaptivní proces pro tento test byl opět naprogramován v jazyce pro numerické výpočty Octave.

### Úloha typu A

Pro různé hodnoty  $\varepsilon_1$  a  $p$  byly sledovány výše uvedené parametry, dále doba potřebná pro adaptaci<sup>56</sup> ( $T$ ) a počet skutečně provedených iterací ( $ITER$ ), než došlo k zastavení algoritmu. V tabulkách 3.20 a 3.21 jsou zaneseny finální hodnoty sledovaných parametrů.

$\varepsilon_1$	$p$	$L2E$	$Q_{norm}$	$ITER$	$T$ [s]
10	10	0.009398	0.166626	30	2.17
100	10	0.009357	0.133231	30	2.17
1000	10	0.010095	0.207438	30	2.18
10	20	0.009080	0.144865	30	2.18
100	20	0.008941	0.106692	30	2.17
1000	20	0.009840	0.172916	30	2.17

Tabulka 3.20: Úloha A. Sledované veličiny pro Jacobiovu metodu.

$\varepsilon_1$	$p$	$L2E$	$Q_{norm}$	$ITER$	$T$ [s]
10	10	0.006730	0.140897	30	2.78
100	10	0.004617	0.053180	30	2.80
1000	10	0.005015	0.079605	30	2.79
10	20	0.006868	0.128420	30	2.81
100	20	0.006545	0.043802	30	2.77
1000	20	0.005031	0.054049	30	2.78

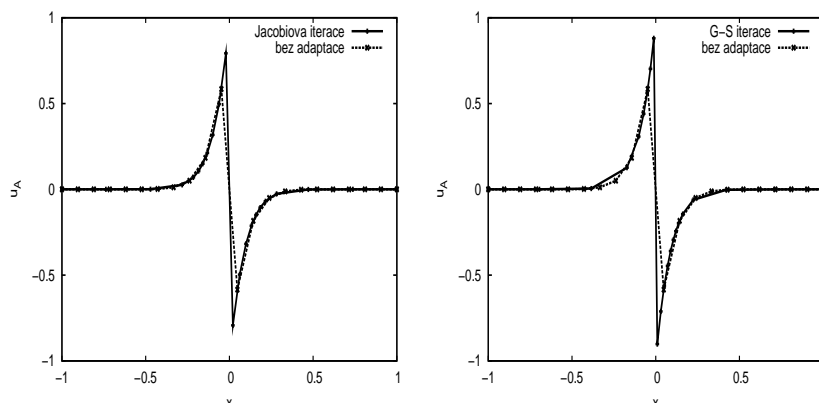
Tabulka 3.21: Úloha A. Sledované veličiny při užití Gaussovy-Seidelovy metody

Porovnání lineární rekonstrukce funkce  $u_A(x)$  na adaptované síti s parametry  $\varepsilon_1 = 100$  a  $p = 20$  pro obě dvě použité metody je zachyceno na obrázku 3.48. Zde je vidět, že lineární rekonstrukce sestavená na síti získané pomocí

<sup>55</sup>Opět byla použita rutina jazyka Octave *quad*.

<sup>56</sup>Byly použity standardní rutiny v Octave. Jedná se o *tic*, *toc*

Gaussovy-Seidelovy metody přesněji zachycuje průběh funkce  $u_A$  na obrázku 3.12.

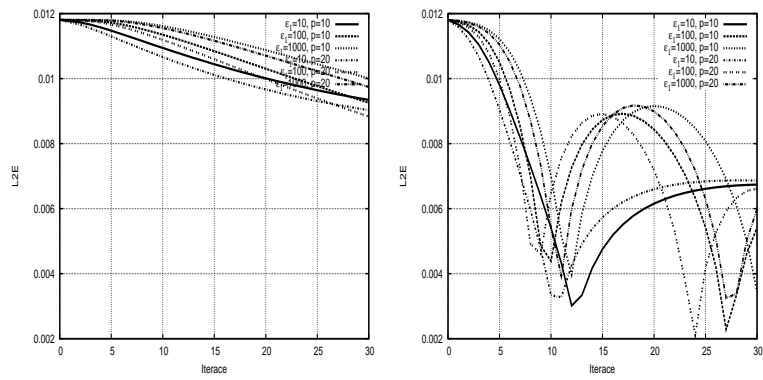


Obrázek 3.48: Úloha A. Porovnání lineární rekonstrukce  $u_h^*$  funkce  $u_A(x)$  na adaptované síti a na síti bez adaptace, při použití Jacobiovy metody (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).

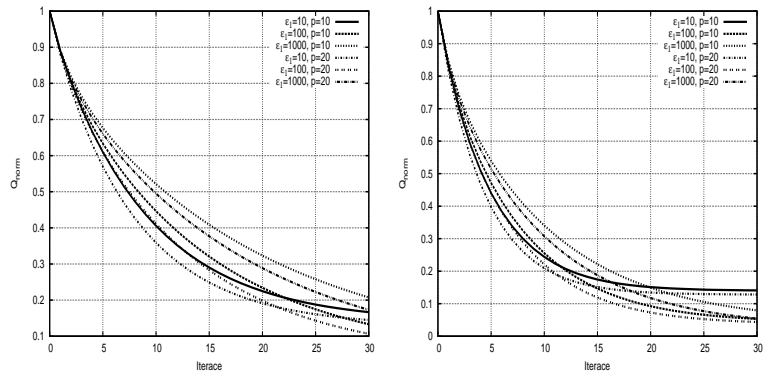
Průběhy sledovaných veličin jsou zachyceny na obrázcích 3.49 a 3.50, zde je vidět, že z hlediska  $Q_{norm}$  obě metody konvergují, rychleji konverguje Gaussova-Seidelova metoda. Z hlediska ukazatele L2E dostáváme lepší výsledky při použití Gaussovy-Seidelovy metody, avšak se zde objevují jisté oscilace. Při větším počtu iterací dochází ke snižování rozkmitu, a tím i k tlumení těchto oscilací, viz zhodnocení provedených experimentů v 1D dále.

Pro stejnou volbu parametrů,  $\varepsilon_1 = 100$  a  $p = 20$  pro obě testované metody, jsou na obrázcích 3.51 zaneseny trajektorie jednotlivých uzlů sítě během adaptačního procesu. Na obrázku 3.51 je vidět, že nedocházelo k přesmýknutí uzlů sítě. Při porovnání trajektorií uzlů sítě pro obě metody je vidět, že Gaussova-Seidelova metoda konverguje k místu nespojitosti (nespojitosť je v bodě nula) rychleji. Porovnání trajektorií při použití Gaussovy-Seidelovy metody<sup>57</sup> pro různé volby parametrů  $\varepsilon_1$  a  $p$  je zaneseno na obrázku 3.52. Zde je vidět, že při zvyšování parametru  $\varepsilon_1$  dochází ke zvětšení poměrů mezi největším a nejmenším objemem sítě, což odpovídá významu tohoto parametru, viz (3.108). Při zvětšování parametru  $p$  při konstantním  $\varepsilon_1$  je síť v okolí nespojitosti více zhuštěná.

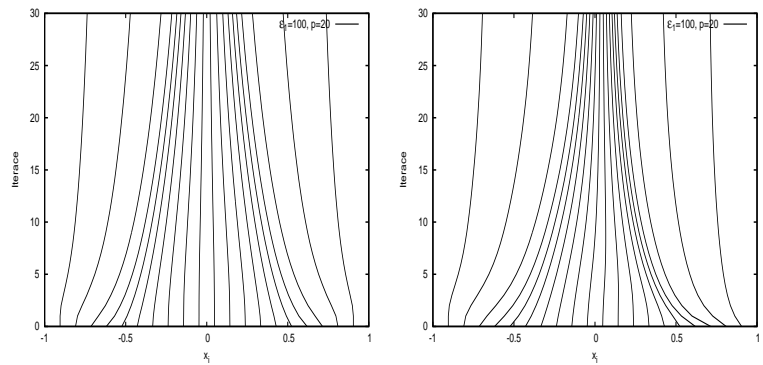
<sup>57</sup>Pro Jacobiovu metodu jsou získané výsledky obdobné.



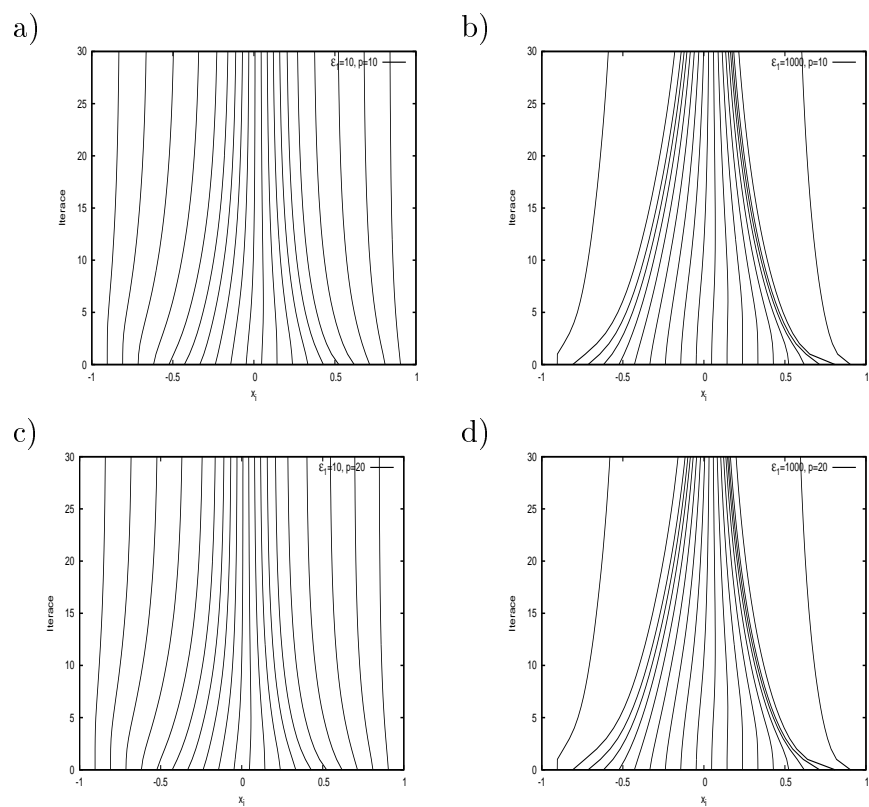
Obrázek 3.49: Úloha A. Průběh  $L_2E$  při použití Jacobiovy metody (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).



Obrázek 3.50: Úloha A. Průběh  $Q_{norm}$  při použití Jacobiovy metody (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).



Obrázek 3.51: Úloha A. Trajektorie jednotlivých uzlů sítě, pro Jacobiovu metodu (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).



Obrázek 3.52: Úloha A. Trajektorie jednotlivých uzlů sítě při použití Gaussovy-Seidelovy metody pro různé volby parametrů  $\varepsilon_1$  a  $p$ , (a)  $\varepsilon_1 = 10, p = 10$ ; (b)  $\varepsilon_1 = 1000, p = 10$ ; (c)  $\varepsilon_1 = 10, p = 20$ ; (d)  $\varepsilon_1 = 1000, p = 20$ .

## Úloha typu B

Postupujeme stejně jako v předchozím a použijme nyní funkci  $u_B$ , viz (3.73). Následující tabulky 3.22 a 3.23 zaznamenávají získané hodnoty. Je zde opět jako v kapitole 3.1.6 vidět nárůst doby potřebné pro adaptaci vůči předchozí úloze, to je způsobeno tím, že v tomto případě používáme pro výpočet ukazatele  $L2E$ , viz (3.147), numerickou integraci, zatímco pro předchozí úlohu je tento ukazatel určen analyticky.

$\varepsilon_1$	$p$	$L2E$	$Q_{norm}$	$ITER$	$T$ [s]
10	10	0.001309	0.270048	30	7.39
100	10	0.001299	0.242325	30	7.37
1000	10	0.001518	0.304269	30	7.40
10	20	0.001265	0.244023	30	7.39
100	20	0.001189	0.206570	30	7.39
1000	20	0.001442	0.272197	30	7.45

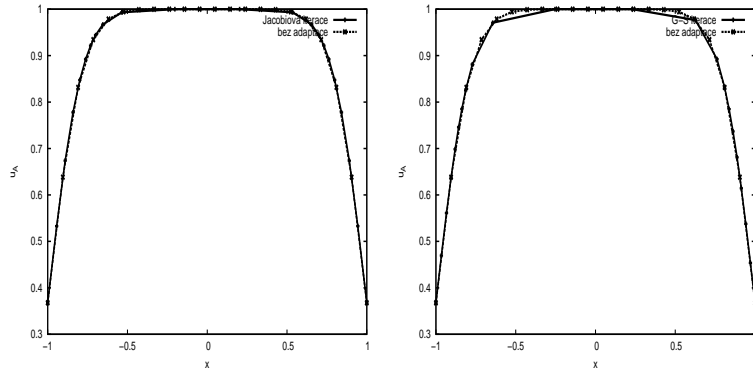
Tabulka 3.22: Úloha B. Sledované veličiny pro Jacobiovu metodu.

$\varepsilon_1$	$p$	$L2E$	$Q_N$	$ITER$	$T$ [s]
10	10	0.001071	0.217919	30	7.50
100	10	0.000900	0.117455	30	7.53
1000	10	0.000985	0.139797	30	7.53
10	20	0.001076	0.203266	30	7.54
100	20	0.000920	0.095470	30	7.54
1000	20	0.000966	0.110163	30	7.49

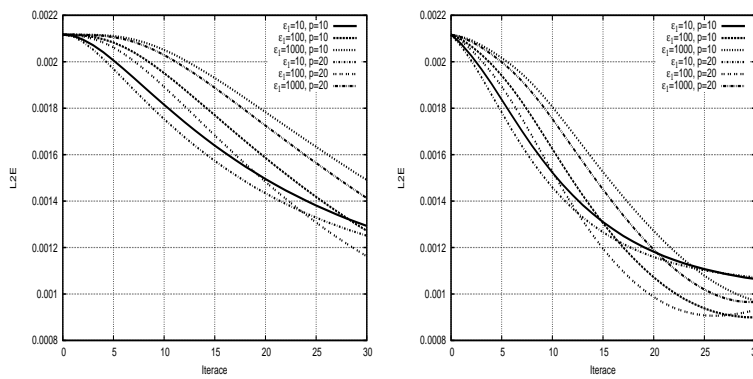
Tabulka 3.23: Úloha B. Sledované veličiny pro Gaussovu-Seidelovu metodu.

Porovnání aproximace funkce  $u_B(x)$  na adaptované síti s parametry  $\varepsilon_1 = 100$  a  $p = 20$  pro obě dvě použité metody je zachyceno na obrázku 3.53. Průběhy sledovaných veličin jsou zachyceny na obrázcích 3.54 a 3.55. Z průběhu  $Q_{norm}$  je vidět, že i v tomto případě konvergovala Gaussova-Seidelova metoda rychleji a z hlediska ukazatele  $L2E$  dosáhla lepších výsledků.

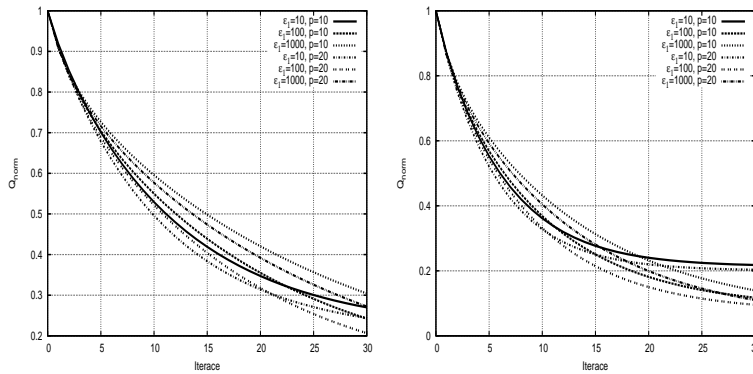
Pro parametry  $\varepsilon_1 = 100$  a  $p = 20$  pro obě testované metody jsou na obrázcích 3.56 zaneseny trajektorie jednotlivých uzlů sítě během adaptačního procesu. Zde je vidět, že nedocházelo k přemýknutí uzlů sítě. Na obrázcích 3.57 jsou porovnány trajektorie vrcholů sítě během adaptačního procesu pro Gaussovu-Seidelovu metodu pro různé volby parametrů  $\varepsilon_1$  a  $p$ . Opět jako v předešlé úloze dochází s růstem parametru  $\varepsilon_1$  k růstu poměru mezi největším a nejmenším objemem sítě. Vliv růstu parametru  $p$  na výslednou síť při stejném  $\varepsilon_1$  byl menší než v předchozí úloze s funkcí  $u_A$ .



Obrázek 3.53: Úloha B. Porovnání lineární rekonstrukce  $u_h^*$  funkce  $u_B(x)$  na adaptované síti a na síti bez adaptace, při použití Jacobiovy metody (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).

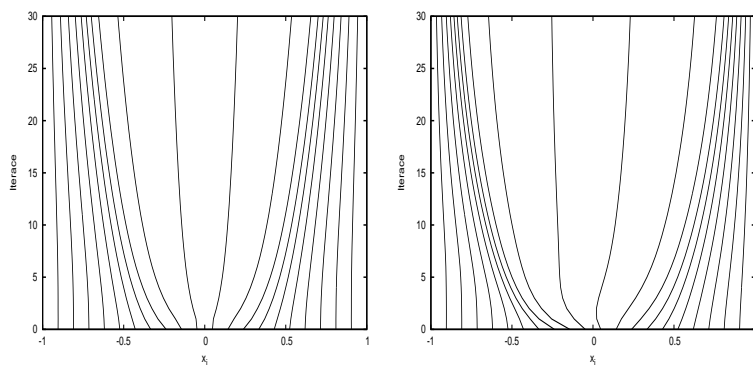


Obrázek 3.54: Úloha B. Průběh L2E při použití Jacobiovy metody (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).

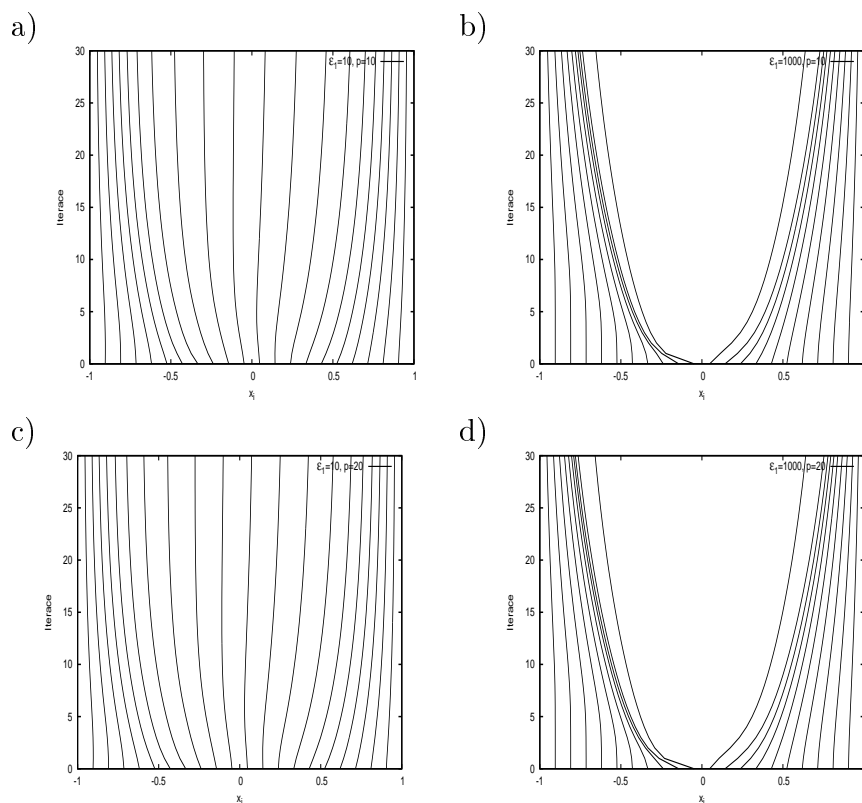


Obrázek 3.55: Úloha B. Průběh  $Q_{norm}$  při použití Jacobiovy metody (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).





Obrázek 3.56: Úloha B. Trajektorie jednotlivých uzlů sítě, pro Jacobiovu metodu (vlevo) a při použití Gaussovy-Seidelovy metody (vpravo).



Obrázek 3.57: Úloha B. Trajektorie jednotlivých uzlů sítě při použití Gaussovy-Seidelovy metody pro různé volby parametrů  $\varepsilon_1$  a  $p$ , (a)  $\varepsilon_1 = 10$ ,  $p = 10$ ; (b)  $\varepsilon_1 = 1000$ ,  $p = 10$ ; (c)  $\varepsilon_1 = 10$ ,  $p = 20$ ; (d)  $\varepsilon_1 = 1000$ ,  $p = 20$ .

### Zhodnocení provedených experimentů v 1D

Z provedených experimentů vyplývá, že obě metody jsou použitelné a výsledná síť závisí na volbě použitých parametrů  $\varepsilon_1$  a  $p$ . Rychleji konverguje postup používající Gaussovu-Seidelovu iterační metodu. U Gaussovy-Seidelovy metody docházelo v první úloze k oscilacím v ukazateli L2E, viz obrázek 3.49. Tyto oscilace je možné utlumit, když použijeme místo Gaussovy-Seidelovy metody metodu SOR<sup>58</sup> s relaxačním parametrem menším než jedna. Poznamenejme, že tyto oscilace nejsou způsobeny oscilacemi uzlů sítě, viz jejich trajektorie na obrázku 3.52, nebo průběh  $Q_{norm}$ , kde je vidět, že metody konvergují. Vznik těchto oscilací je umožněn tím, že naše metody primárně neminimalizují ukazatel L2E, nýbrž funkci  $Q$  (3.133).

Porovnáme-li obdržené výsledky s výsledky získanými pomocí variačního přístupu, viz kapitola 3.1.6, lze vidět, že z hlediska hodnot a průběhu ukazatele L2E, jsou oba přístupy podobné, také zde docházelo k oscilacím daného ukazatele. Z hlediska časové náročnosti vychází pro úlohu A lépe naše metoda (oba postupy), pro úlohu B jsou časy podobné. Za povšimnutí stojí též porovnání trajektorií jednotlivých uzlů. V případě použití našeho přístupu dochází k pohybu většího počtu uzlů, pohybují se i uzly dále od nespojitosti.

### 3.3.6 Numerické experimenty ve 2D

V této části práce otestujeme námi navržené algoritmy pro adaptaci sítě konečných objemů, výsledky zde dosažené jsou původní. Místo lineární rekonstrukce  $\mathbf{w}_h^*$  získané z numerického, po částech konstantního řešení  $\mathbf{w}$ , Eulerových rovnic, použijeme lineární rekonstrukce funkcí  $u_A$  (3.79), resp.  $u_B$  (3.80), získané pomocí vztahu (3.82), kde místo hodnot  $\mathbf{w}_i^k$  použijeme hodnotu funkce  $u_A$ , resp.  $u_B$  v těžišti daného objemu. Pro tento test uvažujeme oblast  $[-5, 5] \times [-5, 5]$ . Dále budeme testovat pohyb hraničních uzlů po nelineární hranici.

Opět jako v 1D budeme hodnotit i vliv parametrů  $\varepsilon_1$  a  $p$  na výslednou síť. Sledované parametry budou<sup>59</sup>  $Q_{norm} = Q_{\mathcal{T}_h} / Q_{\mathcal{T}_h, init}$ , kde  $Q_{\mathcal{T}_h}$  je parametr kvality sítě viz (3.95) a  $Q_{\mathcal{T}_h, init}$  je jeho počáteční hodnota na neadaptované síti. Další sledovaný parametr bude poměr mezi nejdelší a nejkratší hranou sítě, označme jej  $\varepsilon$ , počet skutečně proběhlých globálních iterací  $ITER$  a čas  $T$  nutný pro adaptaci sítě. Hodnotu  $\alpha$  pro podmínku minimálního úhlu, viz (3.111), volíme  $\alpha = 0.02$ . Použitá nastavení jednotlivých metod udávají následující tabulky:

Parametr	Hodnota	Popis
$\beta$	10	multiplikátor váhy účelové funkce
<i>MAXITG</i>	5	počet iterací metody zlatého řezu
<i>MAXITBFGS</i>	1	počet iterací metody BFGS
<i>TSTEP</i>	5	počet iterací nad jedním uzlem
<i>MAXITER</i>	30	počet globálních iterací

Tabulka 3.24: Použité parametry pro metodu s bariérovou funkcí, viz algoritmy 3.2 a 3.3

<sup>58</sup>Successive Overrelaxation

<sup>59</sup>Hodnota parametru kvality sítě je závislá na  $\varepsilon_1$  a  $p$ , z toho důvodu budeme používat normování.

Parametr	Hodnota	Popis
$a$	4	parametr $a$ , viz 3.125
$d$	2	rychlost přechodu $\Psi_{P_i}$ , viz 3.125
$MAXITG$	5	počet iterací metody zlatého řezu
$MAXITBFGS$	1	počet iterací metody BFGS
$MAXITER$	30	počet globálních iterací

Tabulka 3.25: Použité parametry pro metodu s kontrolou úhlů, viz algoritmy 3.3 a 3.4

Parametr	Hodnota	Popis
$\gamma_r$	0.9	multiplikátor, viz 3.126
$MAXITG$	5	počet iterací metody zlatého řezu
$MAXITREP$	5	počet opravných iterací
$MAXITGRAD$	1	počet iterací gradientní metody nad jedním uzlem
$MAXITER$	60	počet globálních iterací

Tabulka 3.26: Použité parametry pro metodu s opravným cyklem, viz algoritmy 3.5, 3.6, 3.7 a 3.8

V průběhu numerických experimentů se ukázalo jako výhodné použít následující zhlazení matice  $\mathbb{M}$ , viz (3.106):

$$\tilde{\mathbb{M}}(P_i) := \frac{1}{|\mathcal{N}(P_i)| + 1} \left( \mathbb{M}(P_i) + \sum_{P_j \in \mathcal{N}(P_i)} \mathbb{M}(P_j) \right), \quad (3.148)$$

kde  $\mathcal{N}(P_i)$  značí množinu sousedních uzlů k uzlu  $P_i$  a  $|\mathcal{N}(P_i)|$  její mohutnost.

### Úloha typu A

Pro tuto úlohu byla vybrána triangulace  $\mathcal{T}_h$  s 4096 objemy a 2136 uzly a s počátečním poměrem  $\varepsilon = 2.38$ . Následující tabulky shrnují sledované parametry.

$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	20	0.464257	5.63	30	11.36
10000	20	0.461356	5.52	30	11.38
1000	40	0.476523	6.45	30	11.13
10000	40	0.469361	6.17	30	11.16

Tabulka 3.27: Úloha typu A. Sledované veličiny pro metodu s bariérovou funkcí.

Ze získaných hodnot je vidět, že použité metody jsou z hlediska parametru  $Q_{norm}$  a  $\varepsilon$  téměř rovnocenné. Z hlediska časové náročnosti je časově nejnáročnější metoda s bariérovou funkcí, která je nejsložitější. Nejlepších výsledků dosáhla metoda s opravným cyklem a to přesto, že počet iterací byl dvojnásobný<sup>60</sup> než

<sup>60</sup>Metoda konverguje pomaleji než předchozí metody, tedy je třeba větší počet iterací, ale doba potřebná pro provedení jedné iterace je kratší.

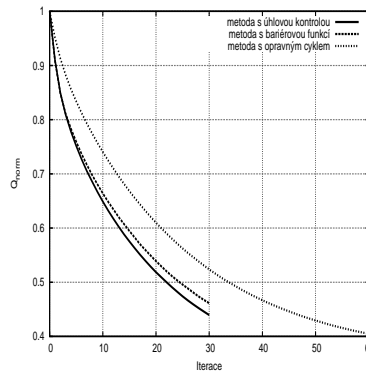
$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	20	0.442497	6.27	30	5.33
10000	20	0.439480	6.01	30	6.09
1000	40	0.449078	6.99	30	5.34
10000	40	0.443641	7.02	30	5.35

Tabulka 3.28: Úloha typu A. Sledované veličiny pro metodu s kontrolou úhlů.

$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	20	0.409717	5.89	60	2.76
10000	20	0.404768	6.04	60	2.75
1000	40	0.422371	6.97	60	2.76
10000	40	0.413969	7.09	60	2.75

Tabulka 3.29: Úloha typu A. Sledované veličiny pro metodu s opravným cyklem.

v předchozích dvou případech. Poznamenejme, že došlo k jistému zkreslení v hodnotách  $Q_{norm}$  pro metodu s bariérovou funkcí a metodu s kontrolou úhlů, které je způsobené tím, že se pohybovaly jen nehraniční uzly sítě. Průběh veličiny  $Q_{norm}$  je téměř identický pro různé volby parametrů  $\varepsilon_1$  a  $p$ , z tohoto důvodu je zanesen jen pro hodnoty  $\varepsilon_1 = 10000$  a  $p = 20$ .

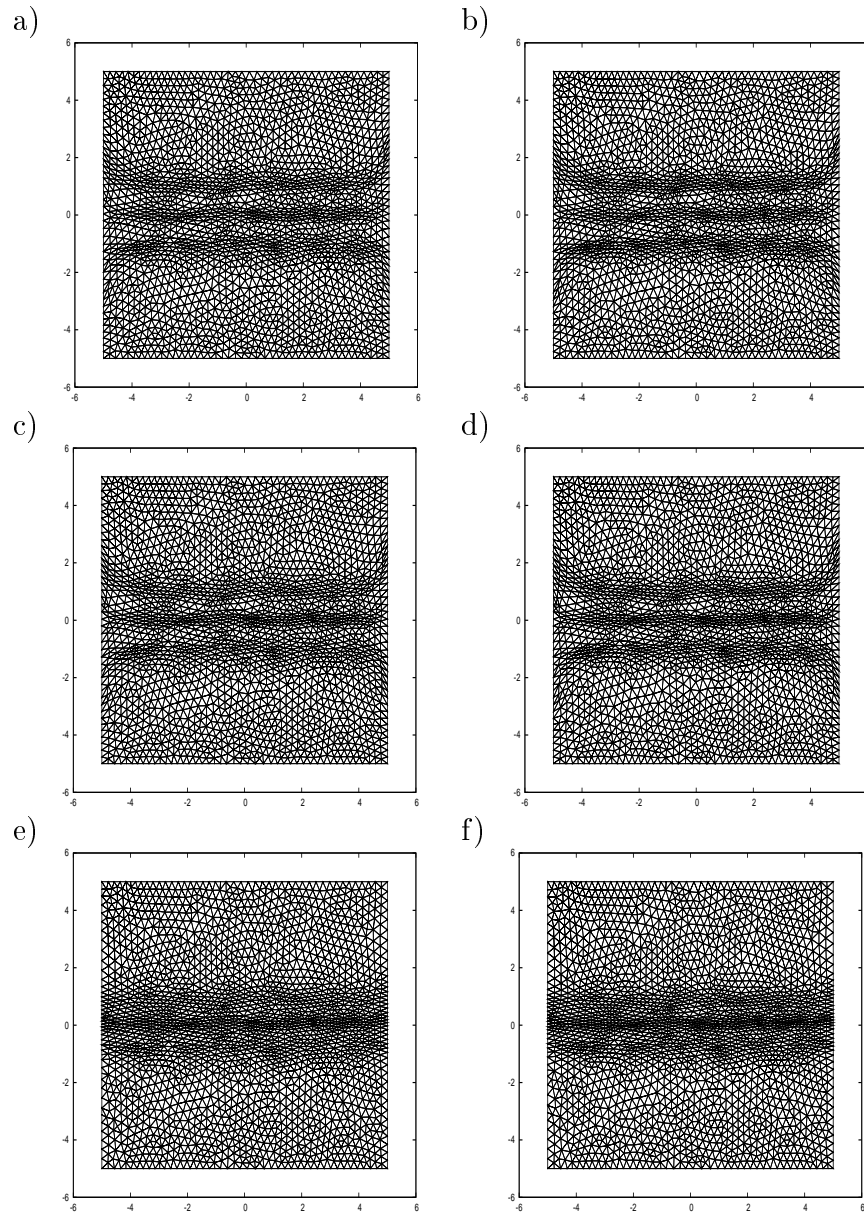


Obrázek 3.58: Úloha typu A. Průběh  $Q_{norm}$  pro testované metody.

Na obrázku 3.59 jsou zaneseny výsledné sítě pro různé hodnoty parametru  $p$ . Je zde vidět, že zdvojnásobení hodnoty  $p$  se na přechodu mezi zjemněnou a nezjemněnou částí sítě projeví pouze nepatrně, což je ostatně vidět i ve výše uvedených tabulkách, hodnota  $\varepsilon$ . Dále je zde vidět, že fixace uzlů na hranici<sup>61</sup> vede k jistým deformacím sítě na hranici.

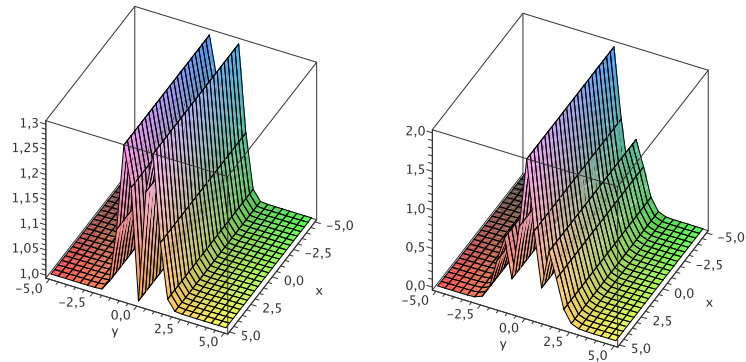
Porovnáme-li sítě na obrázku 3.59 se sítěmi, které jsme obdrželi pomocí variačních metod, viz kapitola 3.1.7 obrázek 3.36, je jasné vidět, že obě metody produkují poněkud rozdílné sítě. Důvodem je to, že v případě variačních metod je adaptační proces závislý na monitorovací funkci  $\mathbb{G}$ , pro jejíž konstrukci používáme gradient funkce  $u_A$  (viz (3.43) a (3.49)), zatímco v tomto případě je adaptační proces závislý na Hessově matici dané funkce, respektive její aproximaci. Na následujícím obrázku 3.60 je pro porovnání zanesena Euklidova norma

<sup>61</sup>Metoda s bariérovou funkcí a s kontrolou úhlů mají fixované uzly na hranici.



Obrázek 3.59: Úloha typu A. Výsledné sítě pro testované metody: (a), (b) metoda s bariérovou funkcí : (a)  $\varepsilon_1 = 10000$ ,  $p = 20$ , (b)  $\varepsilon_1 = 10000$ ,  $p = 40$ , (c), (d) metoda s kontrolou úhlů : (c)  $\varepsilon_1 = 10000$ ,  $p = 20$ , (d)  $\varepsilon_1 = 10000$ ,  $p = 40$ , (e), (f) metoda s opravným cyklem : (e)  $\varepsilon_1 = 10000$ ,  $p = 20$ , (f)  $\varepsilon_1 = 10000$ ,  $p = 40$ .

$\|\nabla u_A\|_2$  gradientu  $u_A$  a Frobeniova norma  $\|\mathbb{H}\|_F$  Hessovy matice funkce  $u_A$ . Při porovnání výše uvedených sítí je vidět, že variační metody s monitorovací funkcí  $\mathbb{G}$  typu ((3.43) a (3.49)) adaptují síť v oblasti s vyšší hodnotou normy gradientu funkce  $u_A$ , zatímco v případě naší metody dochází ke zjemnění v oblasti s vyšší hodnotou normy  $\|\mathbb{H}\|_F$  Hessovy matice funkce  $u_A$ .



Obrázek 3.60: Průběh Euklidovy normy  $\|\nabla u_A\|_2$  gradientu  $u_A$  (vlevo) a Frobeniovy normy  $\|\mathbb{H}\|_F$  Hessovy matice funkce  $u_A$  (vpravo).

## Úloha typu B

Pro úlohu s funkcí  $u_B$  použita počáteční síť  $\mathcal{T}_h$  s 9920 objemy a 5091 uzly, počáteční poměr činil  $\varepsilon = 2.08$ . Následující tabulky 3.30, 3.31 a 3.32 shrnují sledované parametry.

$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	1	0.910077	4.10	30	16.99
10000	1	0.908076	4.19	30	16.92
1000	2	0.919188	4.23	30	15.07
10000	2	0.915307	4.00	30	15.20
1000	20	0.953638	3.47	30	12.88
10000	20	0.934303	3.83	30	12.91
1000	40	0.964699	3.44	30	12.57
10000	40	0.939709	3.73	30	12.68

Tabulka 3.30: Úloha typu B. Sledované veličiny pro metodu s bariérovou funkcí.

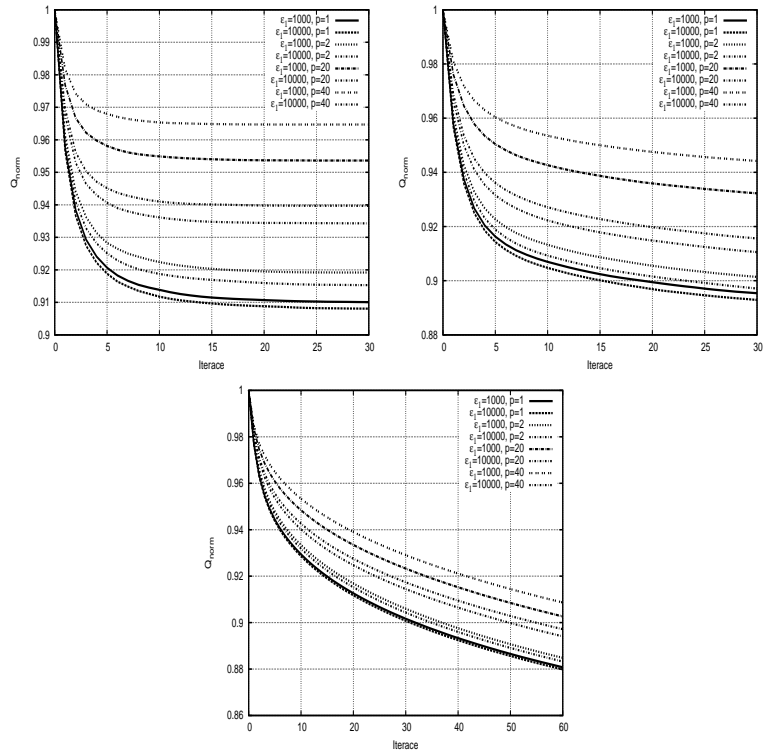
$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	1	0.895382	4.78	30	12.28
10000	1	0.892909	4.69	30	12.30
1000	2	0.901402	5.39	30	12.34
10000	2	0.897102	5.67	30	12.36
1000	20	0.932252	5.46	30	12.41
10000	20	0.913847	6.24	30	12.45
1000	40	0.944207	5.40	30	12.36
10000	40	0.915592	6.89	30	12.44

Tabulka 3.31: Úloha typu B. Sledované veličiny pro metodu s kontrolou úhlů.

$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	1	0.880765	7.19	60	9.13
10000	1	0.879871	7.28	60	9.14
1000	2	0.884822	8.00	60	9.14
10000	2	0.883234	8.23	60	9.14
1000	20	0.902672	8.89	60	9.13
10000	20	0.894084	9.54	60	9.10
1000	40	0.908673	8.47	60	9.14
10000	40	0.897156	9.71	60	9.14

Tabulka 3.32: Úloha typu B. Sledované veličiny pro metodu s opravným cyklem.

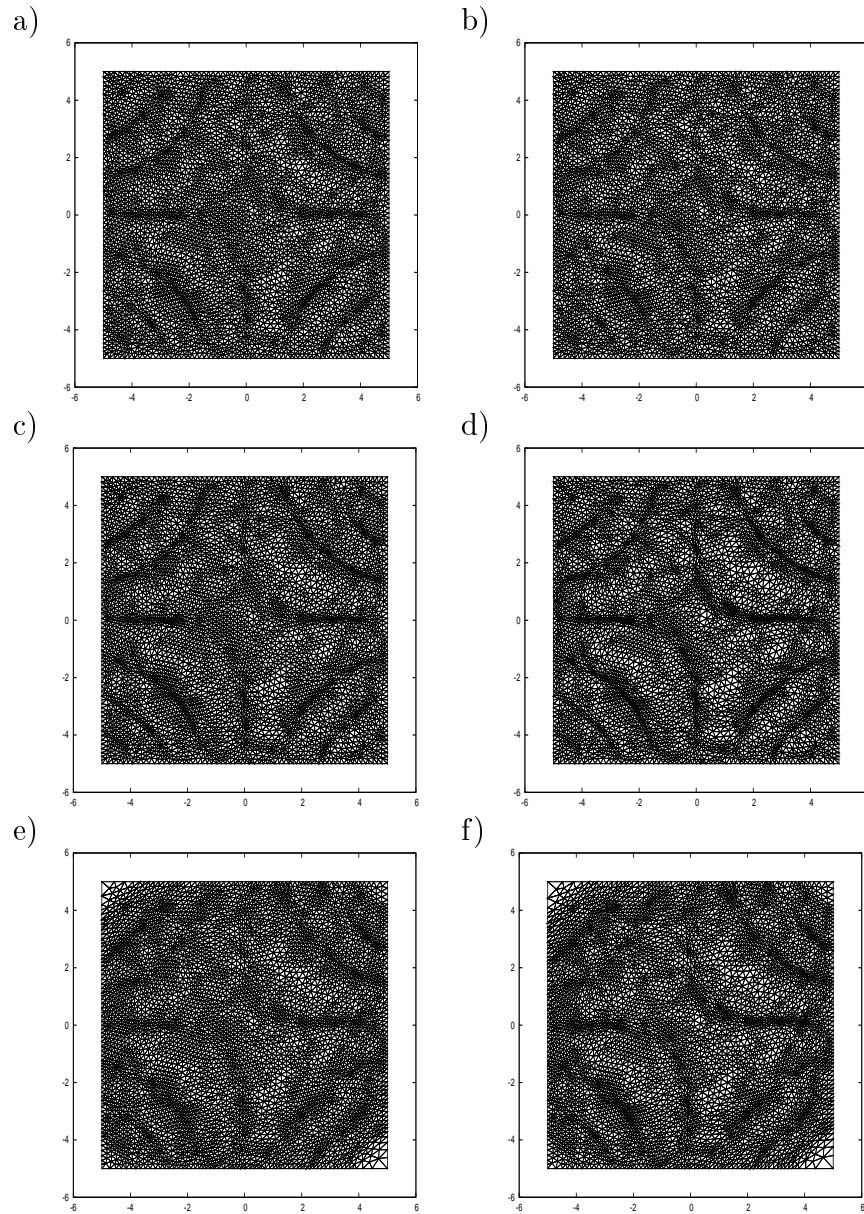
Z naměřených veličin je vidět, že u metody s bariérovou funkcí docházelo ke změně času potřebného pro vykonání adaptace, což je způsobeno tím, že docházelo k přerušení výpočtu ve vnitřním cyklu metody, z důvodu nezlepšení kvality uzlu, viz algoritmus 3.2. Z hlediska časové náročnosti dosáhla nejlepších výsledků metoda s opravným cyklem. Z hlediska kvality sítě  $Q_{norm}$  jsou výsledky metod s kontrolou úhlů a metody s opravným cyklem zhruba stejné.



Obrázek 3.61: Úloha typu B. Průběh  $Q_{norm}$  pro testované metody, metoda s bariérovou funkcí (vlevo), metoda s kontrolou úhlů (vpravo), metoda s opravným cyklem (dole).

Na obrázcích 3.61 je zakreslen průběh veličiny  $Q_{norm}$ . Průběh  $Q_{norm}$  pro metodu s bariérovou funkcí ukazuje, že tato metoda nebyla schopna více snížit tento parametr a tedy lépe adaptovat síť. Výsledné sítě pro různou volbu parametrů jsou zaneseny na obrázku 3.62. Zde je vidět, že při optickém porovnání sítí je nejlépe adaptovaná síť získaná použitím metody s kontrolou úhlů, i když parametr  $\epsilon$ , určující poměr mezi nejdelší a nejkratší hranou sítě, by napovídal, že to bude síť získaná metodou s opravným cyklem, viz tabulka 3.32. Vyšší hodnota  $\epsilon$  pro tuto metodu je způsobena povolením pohybu uzlů po hranici, tím došlo ke zvětšení objemů v rozích oblasti, viz obrázky 3.62 ((e),(f)).

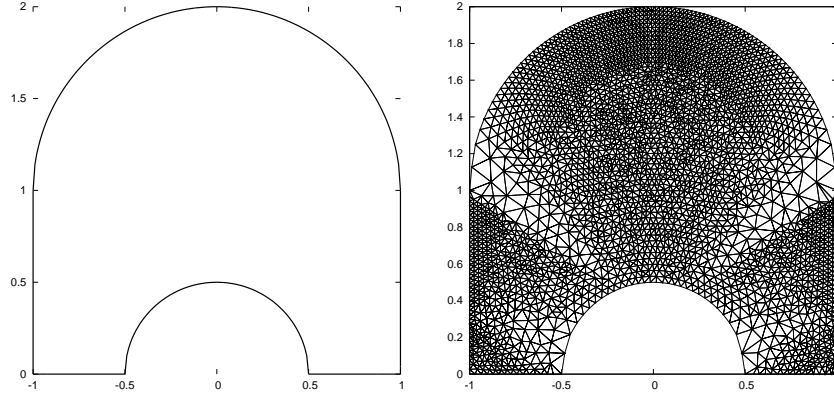




Obrázek 3.62: Úloha typu B. Výsledné sítě pro testované metody: (a), (b) metoda s bariérovou funkcí : (a)  $\varepsilon_1 = 10000$ ,  $p = 1$ , (b)  $\varepsilon_1 = 10000$ ,  $p = 40$ , (c), (d) metoda s kontrolou úhlů : (c)  $\varepsilon_1 = 10000$ ,  $p = 1$ , (d)  $\varepsilon_1 = 10000$ ,  $p = 40$ , (e), (f) metoda s opravným cyklem : (e)  $\varepsilon_1 = 10000$ ,  $p = 1$ , (f)  $\varepsilon_1 = 10000$ ,  $p = 40$ .

## Pohyb po nelineární hranici

Pro otestování pohybu uzlů po nelineární hranici metodou s opravným cyklem uvažujme oblast ve tvaru podkovky, která je zakreslena na obrázku 3.63 (vlevo). Na této oblasti uvažujeme triangulaci s 6616 objemy a 3425 uzly, která je zakreslena na obrázku 3.63 (vpravo), a skutečný počáteční poměr mezi nejdelší a nejkratší hranou sítě  $\varepsilon = 8.08$ . Pro její vygenerování byl použit program FreeFem++.



Obrázek 3.63: Oblast  $\Omega$  (vlevo) a počáteční síť na této oblasti (vpravo).

Hranice této oblasti jsou následující:

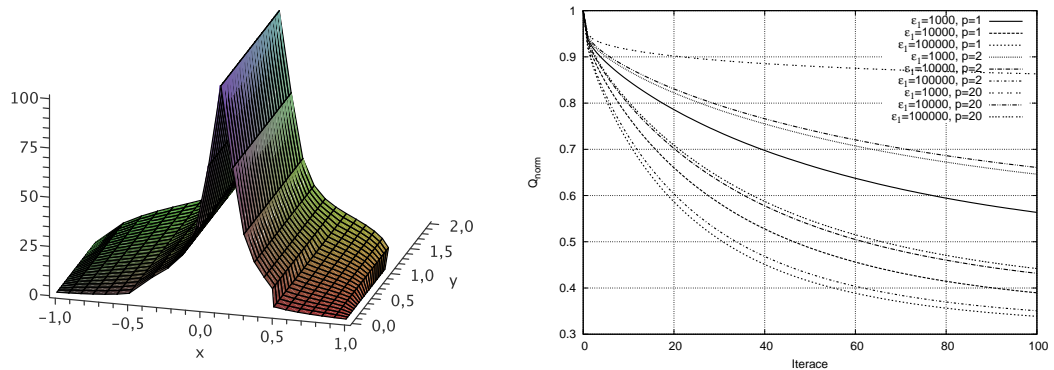
$$\begin{aligned}
 \Gamma_L &= \{ \mathbf{x} \in \mathbb{R}^2; x_1 = -1, x_2 \in (0, 1) \}, \\
 \Gamma_R &= \{ \mathbf{x} \in \mathbb{R}^2; x_1 = 1, x_2 \in (0, 1) \}, \\
 \Gamma_{B_1} &= \{ \mathbf{x} \in \mathbb{R}^2; x_1 \in [-1, -0.5], x_2 = 0 \}, \\
 \Gamma_{B_2} &= \left\{ \mathbf{x} \in \mathbb{R}^2; x_1 \in [-0.5, 0.5], x_2 = \sqrt{0.25 - x_1^2} \right\}, \\
 \Gamma_{B_3} &= \{ \mathbf{x} \in \mathbb{R}^2; x_1 \in [0.5, 1], x_2 = 0 \}, \\
 \Gamma_T &= \left\{ \mathbf{x} \in \mathbb{R}^2; x_1 \in [-1, 1], x_2 = 1 + \sqrt{1 - x_1^2} \right\}.
 \end{aligned} \tag{3.149}$$

Opět jako v předchozích úlohách použijeme místo po částech lineární rekonstrukce  $\mathbf{w}_h^*$ , získané z numerického řešení Eulerových rovnic, po částech lineární rekonstrukci z námi předepsané funkce  $u_C(\mathbf{x})$  (3.150). Tuto rekonstrukci získáme pomocí vztahu (3.82), kde místo hodnot  $\mathbf{w}_i^k$  použijeme hodnotu funkce  $u_C$  v těžišti daného objemu. Zvolili jsme následující funkci  $u_C$ :

$$u_C(\mathbf{x}) = \frac{1}{x_1^2 + 0.01}, \tag{3.150}$$

jejíž průběh je zanesen na obrázku 3.64 (vlevo). Vyjma počtu globálních iterací  $MAXITER = 100$  bylo použito nastavení dané tabulkou 3.26. Důvodem pro zvýšení hodnoty  $MAXITER$  bylo to, abychom získali zřetelnější obrazy sítě pro vizuální porovnání, než při původním nastavení, kde bylo  $MAXITER = 60$ .

Hodnoty sledovaných veličin pro různé volby  $\varepsilon_1$  a  $p$  udává tabulka 3.33, ze získaných hodnot je vidět, že s rostoucí volbou parametru  $\varepsilon_1$  dochází k růstu poměru  $\varepsilon$  mezi nejdelší a nejkratší hranou sítě a dále klesá ukazatel  $Q_{norm}$ , což znamená, že z hlediska tohoto ukazatele, se síť stává více adaptovanou. Průběh



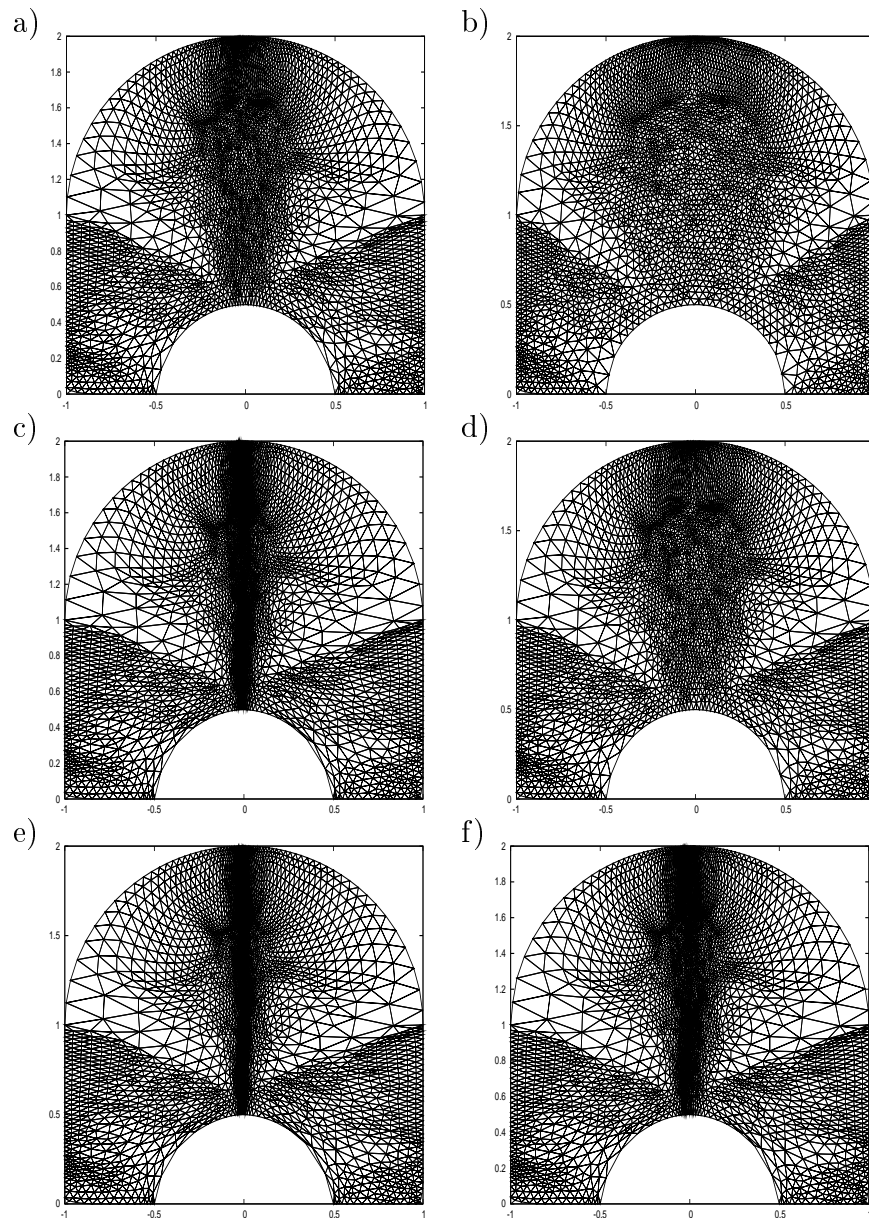
Obrázek 3.64: Průběh funkce  $u_C$  na oblasti  $\Omega$  (vlevo). Průběh  $Q_{norm}$  při užití metody s opravným cyklem pro různé volby parametrů  $\varepsilon_1$  a  $p$  (vpravo).

$Q_{norm}$  je zanesen na obrázku 3.64 (vpravo), zde je vidět, že metoda konvergovala pro všechny volby parametrů.

Porovnání výsledných sítí pro různé volby  $\varepsilon_1$  a  $p$  je zaneseno na obrázku 3.65. Zde je vidět, že ve všech případech došlo ke zjemnění sítě v místě, kde se funkce  $u_C$  prudce mění. Dále je vidět, že s růstem  $\varepsilon_1$  dochází k většímu zjemnění sítě v této oblasti. Je zde patrný i vliv parametru  $p$ . Se vzrůstající hodnotou tohoto parametru je přechod sítě mezi oblastmi, kde se funkce  $u_C$  prudce mění a kde je téměř konstantní, pozvolnější.

$\varepsilon_1$	$p$	$Q_{norm}$	$\varepsilon$	$ITER$	$T$ [s]
1000	1	0.563553	53.46	100	7.40
10000	1	0.389550	154.76	100	7.45
100000	1	0.338696	312.88	100	7.49
1000	2	0.646179	45.01	100	7.37
10000	2	0.431701	118.03	100	7.41
100000	2	0.350720	270.71	100	7.48
1000	20	0.863299	24.43	100	7.36
10000	20	0.660766	46.85	100	7.38
100000	20	0.442039	120.02	100	7.43

Tabulka 3.33: Sledované veličiny pro úlohu s nelineární hranicí, metoda s opravným cyklem.



Obrázek 3.65: Úloha s nelineární hranicí. Výsledné sítě pro metodu s opravným cyklem (a)  $\varepsilon_1 = 1000$ ,  $p = 1$ , (b)  $\varepsilon_1 = 1000$ ,  $p = 20$ , (c)  $\varepsilon_1 = 10000$ ,  $p = 1$ , (d)  $\varepsilon_1 = 10000$ ,  $p = 20$ , (e)  $\varepsilon_1 = 100000$ ,  $p = 1$ , (f)  $\varepsilon_1 = 100000$ ,  $p = 20$ .

## Zhodnocení provedených experimentů ve 2D

Z provedených experimentů vyplývá, že všechny námi navržené metody jsou použitelné pro adaptaci pohybu uzlů sítě. Nejlepších výsledků, jak z hlediska času potřebného pro adaptaci, tak z hlediska parametru  $Q_{norm}$ , dosahovala metoda s opravným cyklem, viz algoritmus 3.8, kde je navíc snadná implementace pohybu uzlů po hranici oblasti. Protože v úlohách, kde byly pozice uzlů fixovány na hranici, docházelo k deformacím sítě, budeme v dalších prováděných testech používat metodu s opravným cyklem. Námi navržená metoda s kontrolou úhlů je schopná zabránit vzniku objemů, jejichž hrany svírají úhel menší, než je předepsaná hodnota  $\delta_{min}$  a je možné ji též použít na “opravu“ sítě, pokud se zde vyskytnou objemy, jejichž hrany svírají úhel menší, než je předepsaná hodnota.

Dále jsme ukázali, že adaptované sítě získané pomocí naší metody se liší od sítí získaných pomocí varičního přístupu, viz provedené experimenty v kapitole 3.1.7. A obdobně jako v případě variačních metod je zde závislost na volbě uživatelem zadaných parametrů, v našem případě  $\varepsilon_1$  a  $p$ .

# 4. Adaptivní algoritmy

V této části naší práce se zaměříme na kompletní adaptivní algoritmy, tj na úlohu, kdy budeme adaptovat výpočetní síť během samotného výpočtu. Motivací je opět zpřesnění výpočtu v oblastech, kde se některá ze složek řešení  $\mathbf{w}$  prudce mění v čase a prostoru. Naším cílem je vyvinout algoritmy, které budou schopny konkurovat tzv. Moving Mesh algoritmům, viz např. [40, 39, 47, 5].

## 4.1 Adaptivní algoritmy

V této části práce budeme vycházet z postupu uvedeného v [40, 39], kde byl pro adaptaci sítě skládající se ze čtyřúhelníku použit variační přístup, a ukážeme si naše vlastní rozšíření tohoto algoritmu.

V dalším předpokládejme dělení časového intervalu  $0 < t_0 < t_1 < \dots$  a jako  $\mathcal{D}_h^k = \{D_i^k\}_{i=1, \dots, J^k}$ , označme síť konečných objemů v  $k$ -tém časovém okamžiku, kde  $\mathcal{D}_h^0$  je počáteční síť. Jako  $\mathbf{w}_{\mathcal{D}_h^k}^k$  si označme po částech konstantní funkci definovanou na  $\Omega_h$ , pro kterou platí, že  $\mathbf{w}_{\mathcal{D}_h^k}^k|_{D_i^k} = \mathbf{w}_i^k$  pro všechny  $i \in J^k$ , kde  $D_i^k$  je vnitřek objemu  $D_i^k$  a  $\mathbf{w}_i^k$  je numerické řešení Eulerových rovnic (1.17) na tomto objemu v čase  $t_k$ , získané pomocí metody konečných objemů. Hodnota  $\mathbf{w}_{\mathcal{D}_h^0}^0$  je dána počáteční podmínkou na počáteční síti  $\mathcal{D}_h^0$ .

Nejprve si představme algoritmus 4.1, uvedený v práci [40], který se skládá z následujících tří kroků, jenž jsou postupně vykonávány v každém časovém kroku  $k$ .

### Algoritmus 4.1. Adaptivní algoritmus

1. **Adaptace sítě:**  $\mathcal{D}_h^{k+1} := \text{Adaptace}(\mathcal{D}_h^k, \mathbf{w}_{\mathcal{D}_h^k}^k)$ .

Na základě numerického řešení  $\mathbf{w}_{\mathcal{D}_h^k}^k$  je jedním z postupů uvedených v kapitole 3 provedena adaptace sítě  $\mathcal{D}_h^k$ , čímž je získána síť  $\mathcal{D}_h^{k+1}$ .

2. **Přepočítání řešení:**  $\tilde{\mathbf{w}}_{\mathcal{D}_h^{k+1}}^k := \text{Transformace}(\mathbf{w}_{\mathcal{D}_h^k}^k, \mathcal{D}_h^k, \mathcal{D}_h^{k+1})$ .

Provedení přepočtu řešení  $\mathbf{w}_{\mathcal{D}_h^k}^k$  ze sítě  $\mathcal{D}_h^k$  na síť  $\mathcal{D}_h^{k+1}$ .

3. **Získání nového řešení:**  $\mathbf{w}_{\mathcal{D}_h^{k+1}}^{k+1} := \text{MKO}(\tilde{\mathbf{w}}_{\mathcal{D}_h^{k+1}}^k, \mathcal{D}_h^{k+1})$ .

Pro získání nového numerického řešení  $\mathbf{w}_{\mathcal{D}_h^{k+1}}^{k+1}$  na adaptované síti  $\mathcal{D}_h^{k+1}$  se použije schéma metody konečných objemů.

Z uvedené posloupnosti kroků algoritmu je vidět, že adaptace sítě vždy zůstává za numerickým řešením, tento nedostatek se snaží odstranit námi navržený postup, kde pro adaptaci sítě v časovém kroku  $k$  používáme odhad (predikci) numerického řešení v časovém kroku  $k + 1$ .

## Algoritmus 4.2. Algoritmus s predikcí

1. **Predikce:**  $\mathbf{w}_{\mathcal{D}_h^k}^{k+1} := \text{MKO} \left( \mathbf{w}_{\mathcal{D}_h^k}^k, \mathcal{D}_h^k \right)$ .

Metodou konečných objemů se spočte řešení  $\mathbf{w}_{\mathcal{D}_h^k}^{k+1}$ , které se v dalším kroku použije pro adaptaci.

2. **Adaptace sítě:**  $\mathcal{D}_h^{k+1} := \text{Adaptace} \left( \mathcal{D}_h^k, \mathbf{w}_{\mathcal{D}_h^k}^{k+1} \right)$ .

Na základě spočteného numerického řešení  $\mathbf{w}_{\mathcal{D}_h^k}^{k+1}$  je jedním z postupů uvedených v kapitole 3 provedena adaptace sítě.

3. **Přepočtení řešení:**  $\tilde{\mathbf{w}}_{\mathcal{D}_h^{k+1}}^k := \text{Transformace} \left( \mathbf{w}_{\mathcal{D}_h^k}^k, \mathcal{D}_h^k, \mathcal{D}_h^{k+1} \right)$ .

Provedení přepočtu řešení  $\mathbf{w}_{\mathcal{D}_h^k}^k$  ze sítě  $\mathcal{D}_h^k$  na síť  $\mathcal{D}_h^{k+1}$ .

4. **Získání nového řešení:**  $\mathbf{w}_{\mathcal{D}_h^{k+1}}^{k+1} := \text{MKO} \left( \tilde{\mathbf{w}}_{\mathcal{D}_h^{k+1}}^k, \mathcal{D}_h^{k+1} \right)$ .

Metodou konečných objemů se spočte nové numerické řešení  $\mathbf{w}_{\mathcal{D}_h^{k+1}}^{k+1}$  na adaptované síti  $\mathcal{D}_h^{k+1}$ .

Zde je vidět, že v jednom časovém kroku sítě dochází k dvojnásobnému vyhodnocení schématu metody konečných objemů. Toto vyhodnocení je obecně časově náročné a vede ke zpomalení algoritmu. Postup, který výše uvedené zpomalení částečně eliminuje, je zmíněn v následující poznámce.

**Poznámka k predikci a adaptaci sítě** Z důvodů urychlení výpočtů je predikce a následná adaptace provedena v každém kroku  $k$ , který splňuje podmínku:  $k \bmod s = 0$ , kde  $s \in \mathbb{Z}^+$  je uživatelem předepsaný parametr. Není-li podmínka splněna, platí  $\mathcal{D}_h^{k+1} = \mathcal{D}_h^k$  a neprovádí se ani transformace řešení. Obdobnou podmínku je možné aplikovat i v algoritmu 4.1. Výše uvedené algoritmy lze, z důvodů urychlení výpočtů, upravit také tak, že predikce, resp. adaptace a následný přepočtení hodnot se provádí pouze v některých časových okamžicích  $t_k$ .

**Poznámka k transformaci řešení** Poznamenejme, že pokud schéma transformace řešení ze sítě  $\mathcal{D}_h^k$  na adaptovanou síť  $\mathcal{D}_h^{k+1}$  požaduje, aby docházelo pouze k malému posunu vrcholů sítě, viz následující kapitola 4.2, není možné<sup>1</sup> síť nejprve adaptovat a pak řešení pouze přepočíst. Protože námi používané metody adaptace sítě<sup>2</sup> viz kapitola 3, jsou iterační a při jedné iteraci byl posun vrcholů sítě minimální, budeme transformaci řešení provádět po každém pohybu všech vrcholů sítě, tj. po každé iteraci adaptivního algoritmu, nikoliv až na finální adaptované síti. Je vidět, že dochází ke spárování kroku adaptace sítě s přepočtením řešení.

Oba uvedené algoritmy 4.1 a 4.2 používají explicitní adaptivitu v prostoru, danou adaptací výpočetní sítě. Se změnou prostorového kroku dochází, díky CFL podmínce použitých numerických řešičů, viz kapitola 2.6, též ke změně časového kroku.

<sup>1</sup>Při tomto postupu by nebylo zaručeno, že je posun vrcholů sítě malý.

<sup>2</sup>Zde máme na mysli metody používající variační přístup a metody založené na minimalizaci kvality vrcholu sítě.

## 4.2 Transformace řešení

Z výše uvedených algoritmů plyne, že po adaptaci sítě následuje transformace řešení. Důvodem k tomuto kroku je to, že při adaptaci dojde ke změně poloh vrcholů sítě, následně i objemů sítě a, v případě některých metod<sup>3</sup>, i ke změně topologie sítě. Je proto nutné transformovat řešení  $\mathbf{w}_{\mathcal{D}_h^k}^k$  na řešení  $\tilde{\mathbf{w}}_{\mathcal{D}_h^{k+1}}^k$  tak, aby odpovídalo nové síti. Zde je nutné dodržet to, aby daná transformace nezpůsobila změnu fyzikálního chování daného systému, zejména nám jde o splnění zákonů zachování, které jsou pro úlohy hyperbolického typu, jakými jsou Eulerovy rovnice, klíčové [40, 47, 33]. V případě nedodržení tohoto principu bude už pouhá změna výpočetní sítě způsobovat nárůst, či pokles jedné ze složek  $\mathbf{w}$ , tedy např. hmotnosti.

Abychom se vyhnuli tomuto problému, budeme od každé transformace řešení požadovat, aby byl splněný tzv. *geometrický zákon zachování* (GMCL-Geometric Mass Conservation Law):

$$\sum_{i \in J} |D_i^k| \mathbf{w}_i^k = \sum_{i \in J} |\tilde{D}_i^{k+1}| \tilde{\mathbf{w}}_i^k, \quad (4.1)$$

kde  $J$  představuje indexní množinu,  $|D_i^k|$  představuje  $N$ -dimensionální míru objemu  $D_i^k$  a  $\mathbf{w}_i^k$  představuje po částech konstantní numerické řešení Eulerových rovnic na daném objemu v  $k$ -tém časovém kroku. Odpovídající ekvivalenty na nové adaptované síti budeme označovat vlnovkou  $\tilde{\phantom{x}}$ .

V případech, kde používáme spárování, viz poznámka k transformaci řešení, adaptace sítě a transformace řešení, je nutné dodržet vztah (4.1) i mezi jednotlivými iteracemi algoritmu adaptace<sup>4</sup>, kde horní index  $k$  nyní představuje iteraci algoritmu adaptace sítě, nikoliv časový krok výpočtu. Z tohoto důvodu budeme, pokud to neohrozí srozumitelnost, horní index  $k$  v transformačních vztazích vynechávat.

### 4.2.1 Transformace řešení v 1D

Zabývejme se nejprve úlohou transformace řešení na 1D síti. Použijme opět značení stejné jako v dřívějších kapitolách. V každém časovém okamžiku  $t_k$  mějme dány polohy uzlů sítě  $x_i^k$ ,  $i = 1, \dots, J + 1$ , kde  $J$  je počet objemů sítě a každý objem  $D_i^k$  je tvořen uzly  $x_i^k$  a  $x_{i+1}^k$ , kde platí  $x_i^k < x_{i+1}^k$ . Těžiště objemu  $D_i^k$  označme jako  $x_{i+\frac{1}{2}}^k$  a numerické řešení na tomto objemu jako  $\mathbf{w}_i^k$ . Získáme-li po posunutí novou pozici  $x_{i+\frac{1}{2}}^{k+1}$ , kde  $x_{i+\frac{1}{2}}^{k+1} \in [x_{j-\frac{1}{2}}^k, x_{j+\frac{1}{2}}^k]$ , nabízí se použít pro výpočet transformovaného řešení  $\tilde{\mathbf{w}}_i^k$  lineární interpolaci, viz obrázek 4.1:

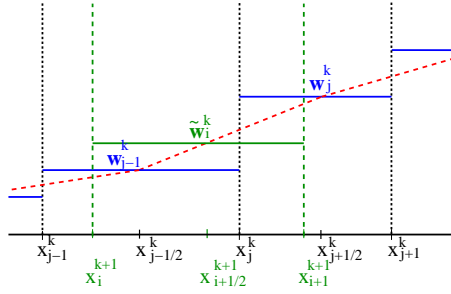
$$\tilde{\mathbf{w}}_i^k = \frac{\mathbf{w}_j^k - \mathbf{w}_{j-1}^k}{x_{j+\frac{1}{2}}^k - x_{j-\frac{1}{2}}^k} \left( x_{i+\frac{1}{2}}^{k+1} - x_{j+\frac{1}{2}}^k \right) + \mathbf{w}_j^k.$$

<sup>3</sup>Např. anisotropní adaptivity.

<sup>4</sup>V podstatě jde o to, že v  $k$ -tém časovém kroku mezi situacemi, kdy máme k dispozici síť  $\mathcal{D}_h^k$  a okamžikem, kdy získáme síť  $\mathcal{D}_h^{k+1}$ , se vykoná předepsaný počet iterací algoritmu adaptace sítě spolu s přepočtem řešení a v každé z těchto iterací je též nutné splnit zákon zachování. Poznamenejme, že pokud je geometrický zákon zachování splněn mezi jednotlivými iteracemi algoritmu adaptace sítě, pak je splněn i mezi řešením v první a řešením v poslední iteraci algoritmu adaptace sítě.



V práci [40] se ukazuje, že tato interpolace nesplňuje geometrický zákon zachování a proto je použit následující postup pro získání transformační formule.



Obrázek 4.1: Lineární interpolace jedné ze složek numerického řešení mezi sítě v čase  $t_k$  (černá barva) a adaptovanou sítí v čase  $t_{k+1}$  (zelená barva). Pomocí červené šrafované čáry je naznačena konstrukce řešení  $\tilde{\mathbf{w}}_i^k$  v bodě  $x_{i+\frac{1}{2}}^{k+1}$ .

Předpokládejme, že posunů uzlů  $c(x) = x - \tilde{x}$  je malé<sup>5</sup>,  $|c(x)| \ll 1$ , kde pomocí  $\tilde{x}$  je označena nově spočtená poloha. Dále použijme perturbační metody [40] na následující integrál:

$$\begin{aligned}
\int_{\tilde{x}_i}^{\tilde{x}_{i+1}} \mathbf{w}(\tilde{x}) d\tilde{x} &= \int_{x_i}^{x_{i+1}} \mathbf{w}(x - c(x)) (1 - c_x(x)) dx \\
&\approx \int_{x_i}^{x_{i+1}} (\mathbf{w}(x) - c(x)\mathbf{w}_x(x)) (1 - c_x(x)) dx \\
&\approx \int_{x_i}^{x_{i+1}} (\mathbf{w}(x) - (c\mathbf{w})_x(x)) dx \\
&= \int_{x_i}^{x_{i+1}} \mathbf{w}(x) dx - ((c\mathbf{w})_{i+1} - (c\mathbf{w})_i),
\end{aligned} \tag{4.2}$$

kde jsme zanedbali členy vyšších řádů, pomocí  $(c\mathbf{w})_i$  značíme hodnotu  $(c\mathbf{w})(x)$  na hranici objemu  $D_i$  v uzlu  $x_i$ . Výše uvedený postup vede k následujícímu vztahu (4.3), který nám umožňuje výpočet  $\tilde{\mathbf{w}}_i$ :

$$|\tilde{D}_i| \tilde{\mathbf{w}}_i = |D_i| \mathbf{w}_i - ((c\mathbf{w})_{i+1} - (c\mathbf{w})_i), \tag{4.3}$$

kde tato formule splňuje geometrický zákon zachování (4.1). Pro výpočet hodnoty toku  $(c\mathbf{w})_i$  se použije vhodný numerický upwinding tok, příkladem může být následující tok prvního řádu (*Steger-Warming*):

$$(c\mathbf{w})_i := \frac{c_i}{2} (\mathbf{w}_i + \mathbf{w}_{i-1}) - \frac{|c_i|}{2} (\mathbf{w}_i - \mathbf{w}_{i-1}). \tag{4.4}$$

Nebo numerický tok druhého řádu [40]:

$$(c\mathbf{w})_i := \frac{c_i}{2} (\mathbf{w}_i^+ + \mathbf{w}_i^-) - \frac{|c_i|}{2} (\mathbf{w}_i^+ - \mathbf{w}_i^-), \tag{4.5}$$

kde vektory  $\mathbf{w}_i^+$  a  $\mathbf{w}_i^-$  představují hodnoty lineární rekonstrukce numerického řešení v uzlu sítě  $x_i$  (podobně jako v kapitole 2.6.2, viz obrázek 2.7), jednotlivé

<sup>5</sup>A navíc nedojde k přesmyknutí uzlů sítě, což všechny námi použité metody splňují.

složky  $w_{i,j}^\pm$ ,  $j = 1, \dots, N + 2$  vektoru  $\mathbf{w}_i^\pm$  jsou zkonstruovány:

$$w_{i,j}^+ := w_{i,j} + \frac{1}{2} (x_i - x_{i+1}) S_{i+\frac{1}{2},j}, \quad (4.6)$$

$$w_{i,j}^- := w_{i-1,j} + \frac{1}{2} (x_i - x_{i-1}) S_{i-\frac{1}{2},j}, \quad (4.7)$$

kde  $w_{i,j}$  je  $j$ -tá složka  $\mathbf{w}_i^k$ ,  $S_{i\pm\frac{1}{2},j}$  je aproximace směrnice  $\frac{\partial w_j}{\partial x}$  v bodě  $x_{i\pm\frac{1}{2}}$ . Aproximace  $S_{i+\frac{1}{2},j}$  používá limiter typu (2.75):

$$S_{i+\frac{1}{2},j} := \left( \text{sign}(S_{i+\frac{1}{2},j}^+) + \text{sign}(S_{i+\frac{1}{2},j}^-) \right) \frac{|S_{i+\frac{1}{2},j}^+ S_{i+\frac{1}{2},j}^-|}{|S_{i+\frac{1}{2},j}^+| + |S_{i+\frac{1}{2},j}^-|}, \quad (4.8)$$

kde  $S_{i+\frac{1}{2},j}^+$ , resp  $S_{i+\frac{1}{2},j}^-$ , představují aproximace hodnoty  $\frac{\partial w_j}{\partial x}$  v bodě  $x_{i+\frac{1}{2}}$  zkonstruované pomocí dopředné, resp. zpětné diference:

$$S_{i+\frac{1}{2},j}^+ := \frac{w_{i+\frac{3}{2},j} - w_{i+\frac{1}{2},j}}{x_{i+\frac{3}{2}} - x_{i+\frac{1}{2}}}, \quad S_{i+\frac{1}{2},j}^- := \frac{w_{i+\frac{1}{2},j} - w_{i-\frac{1}{2},j}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}. \quad (4.9)$$

Hodnota aproximace  $S_{i-\frac{1}{2},j}$  je určena analogicky. Použití tohoto postupu vyžaduje po každém posunu uzlů přepočítání hodnoty numerického řešení z předchozí sítě na síť novou. Hodnota  $\tilde{\mathbf{w}}_{\mathcal{D}_h^{k+1}}^k$  je hodnota, kterou získáme přepočtem dle (4.3) po posledním posunutí.

Požadavek malého posunutí uzlů a nutnost po každém posunu uzlů řešení transformovat je značně omezující. Použijme proto náš vlastní přístup, kdy nám stačí znát výpočetní síť a řešení na ní před adaptací a síť po adaptaci. Pro přepočítání řešení použijme váženého průměru:

$$\tilde{\mathbf{w}}_i^k := \frac{\sum_{j \in \mathcal{I}_i^{k+1}} \mathbf{w}_j^k a_{ij}}{\sum_{j \in \mathcal{I}_i^{k+1}} a_{ij}}, \quad (4.10)$$

kde  $\mathcal{I}_i^{k+1}$  značí množinu indexů objemů  $D_j^k \in \mathcal{D}_h^k$ , takových, které nají průnik s objemem  $D_i^{k+1} \in \mathcal{D}_h^{k+1}$  a  $a_{ij}$  označuje  $N$ -dimensionální míru tohoto průniku. Poznamenejme, že v případě, že se nezmění aproximace hranice výpočetní oblasti, což v 1D neuvažujeme<sup>6</sup>, tak (4.10) splňuje geometrický zákon zachování. Tento postup je snadno zobecnitelný do vyšších dimenzí, umožňuje přepočítat řešení i v případě, že došlo ke změně počtu objemů sítě a v 1D je jeho implementace snadná a velmi rychlá.

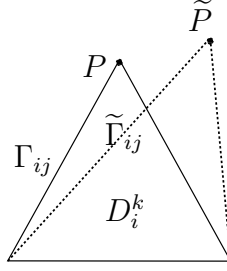
## 4.2.2 Transformace řešení ve 2D

V této části se budeme věnovat transformaci numerického řešení ve 2D. Vyjděme z postupu uvedeného v předchozí kapitole 4.2.1 a upravme ho pro trojúhelníkovou síť  $\mathcal{T}_h$ , jde o 2D modifikaci námi publikovaného postupu [16] pro 3D.

Pomocí posunu vrcholu  $P$  objemu  $D_i^k$  na novou pozici  $\tilde{P}$  definujme na  $D_i^k$  lineární zobrazení  $\mathbf{c}$ , viz obrázek 4.2. Libovolný bod  $x \in D_i^k$  je pak transformován na bod  $\tilde{x} \in D_i^{k+1}$  pomocí:

$$\tilde{x} = x - \mathbf{c}(x), \quad (4.11)$$

<sup>6</sup>Ve 2D dochází při posunu hraničních uzlů po nelineární hranici ke změně polygonální oblasti  $\Omega_h$ .



Obrázek 4.2: Posunutí  $P \rightarrow \tilde{P}$ ,  $\Gamma_{ij} \rightarrow \tilde{\Gamma}_{ij}$

s následujícím Jacobiánem transformace:

$$J(x) = \det \frac{D\tilde{x}}{Dx} = \det \begin{pmatrix} 1 - \frac{\partial c_1}{\partial x_1}, & -\frac{\partial c_1}{\partial x_2} \\ -\frac{\partial c_2}{\partial x_1}, & 1 - \frac{\partial c_2}{\partial x_2} \end{pmatrix}. \quad (4.12)$$

Za předpokladu, že posunutí  $\mathbf{c}$  je malé, lze psát:

$$\begin{aligned} \int_{D_i^{k+1}} \mathbf{w}(\tilde{x}) d\tilde{x} &= \int_{D_i^k} \mathbf{w}(x - \mathbf{c}(x)) |J(x)| dx \\ &= \int_{D_i^k} \mathbf{w}(x - \mathbf{c}(x)) (1 - \operatorname{div} \mathbf{c}(x) + \mathcal{O}) dx \\ &= \int_{D_i^k} (\mathbf{w} - \nabla \mathbf{w} \cdot \mathbf{c} + \mathcal{O}) (1 - \operatorname{div} \mathbf{c}(x) + \mathcal{O}) dx \\ &= \int_{D_i^k} (\mathbf{w} - \mathbf{w} \operatorname{div} \mathbf{c} - \nabla \mathbf{w} \cdot \mathbf{c} + (\nabla \mathbf{w} \cdot \mathbf{c}) \operatorname{div} \mathbf{c} + \mathcal{O}) dx \\ &\approx \int_{D_i^k} (\mathbf{w} - \operatorname{div}(\mathbf{w}\mathbf{c})) dx \\ &= \int_{D_i^k} \mathbf{w} dx - \int_{\partial D_i^k} \mathbf{w} c_n dS, \end{aligned} \quad (4.13)$$

kde  $\mathcal{O}$  značí členy vyšších řádů, které jsou spolu s členem  $(\nabla \mathbf{w} \cdot \mathbf{c}) \operatorname{div} \mathbf{c}$  zanedbány a  $c_n = \mathbf{c} \cdot \mathbf{n}$ , kde  $\mathbf{n}$  je jednotková vnější normála k  $\partial D_i^k$ . Tento postup vede k následujícímu vztahu pro přepočtení řešení:

$$|D_i^{k+1}| \tilde{\mathbf{w}}_i^k = |D_i^k| \mathbf{w}_i^k - \sum_{j \in s(i)} |\Gamma_{ij}| \left( c_{n_{ij}}^+ \mathbf{w}_i^k + c_{n_{ij}}^- \mathbf{w}_j^k \right), \quad (4.14)$$

kde  $s(i)$  je množina indexů objemů sousedících s  $D_i^k$ ,  $\Gamma_{ij} = \partial D_i^k \cap \partial D_j^k$ ,  $|\Gamma_{ij}|$  je  $N$ -dimensionální míra  $\Gamma_{ij}$ ,  $\mathbf{n}_{ij}$  označuje jednotkovou vnější normálu k  $\partial D_i^k$  na  $\Gamma_{ij}$  a pomocí  $\pm$  je označena pozitivní ( $\geq$ ), resp. negativní ( $\leq$ ) část příslušné skalární veličiny. Poznamenejme, že konstanta  $c_{n_{ij}}$  v (4.14) je brána v těžišti  $\Gamma_{ij}$ . Protože použití předchozího vztahu (4.14) předpokládá malý posun vrcholů sítě, je nutné při jeho aplikaci použít dříve zmíněné spárování, viz poznámka k transformaci řešení výše, adaptačního kroku s transformací řešení, tj. vztah (4.14) je třeba vyhodnocovat po každé změně polohy vrcholu, tedy v jednotlivých iteracích adaptivních metod.

Jako alternativu uvažujme způsob přepočtu daný (4.10), toto schéma v případě posunu hraničních uzlů po nelineární části hranice nespĺňuje zákon zachování<sup>7</sup>.

<sup>7</sup>Důvodem je to, že na nové síti mohou existovat takové objemy  $D_i^{k+1} \in \mathcal{T}_h^{k+1}$ , kde  $\sum_{j \in \mathcal{I}_i^{k+1}} a_{ij} \neq |D_i^{k+1}|$ .

Jako další jistou nevýhodu schématu je nutné považovat vyšší časovou náročnost, která je daná nutností výpočtu plochy  $a_{ij}$ ,  $j \in \mathcal{I}_i^{k+1}$  jednotlivých průniků.

### 4.3 Numerické experimenty v 1D

V této části naší práce se zaměříme na otestování výše uvedených algoritmů 4.1 a 4.2 na úlohách, které si zde blíže popíšeme. Jako numerické řešiče budeme používat řešiče popsané v kapitole 2.6 a jako adaptivní modul uvedených algoritmů použijeme, buď variačního přístupu, viz kapitola 3.1.4, nebo naše původní metody založené na kvalitě vrcholu, popsané v kapitole 3.3.4. Pro přepočtení řešení je využito obou postupů zmíněných v kapitole 4.2.1. Pro implementaci všech výše uvedených metod bylo použito jazyka pro numerické výpočty Octave.

#### Burgersova rovnice

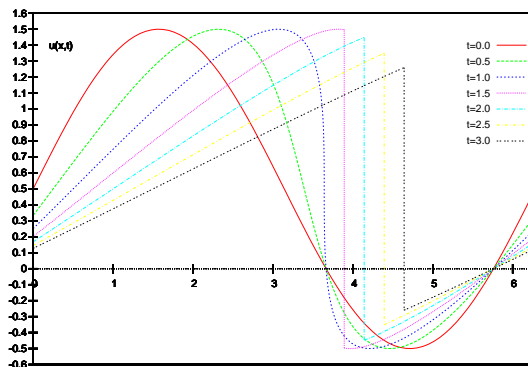
Pro základní ověření funkčnosti adaptivních metod jsme si vybrali nevazkou Burgersovu rovnici [40, 30], která představuje zjednodušení systému rovnic Eulerových (1.2):

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0, \quad 0 \leq x \leq 2\pi. \quad (4.15)$$

Počáteční podmínku pro naši úlohu jsme volili:

$$u^0(x) = \frac{1}{2} + \sin(x), \quad 0 \leq x \leq 2\pi, \quad (4.16)$$

a jako okrajovou podmínku jsme zvolili periodickou okrajovou podmínku. Takto zvolená soustava počátečních a okrajových podmínek pro Burgersovu rovnici vede k řešení<sup>8</sup>, ve kterém se od kritického času  $t_c = 1$  objeví nespojitost, rázová vlna, která se šíří směrem doprava, viz obrázek 4.3.



Obrázek 4.3: Evoluce přesného řešení Burgersovy rovnice.

Otestovali jsme možné kombinace jednotlivých adaptačních technik spolu s metodami pro přepočtení řešení ze staré sítě na novou, přehled testovaných metod udává tabulka 4.1.

Sledované parametry jsou počet proběhlých iterací algoritmu *ITER*, ukazatel chyby L2E, viz (3.75) a doba potřebná pro výpočet *T*. Jako přesné řešení *u*

<sup>8</sup>Uvažujeme slabé po částech hladké řešení ve smyslu [13][Definice 2.12, str. 70].

Značení	Popis
$(P)ATMOVE - PM$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použito variačního přístupu s umělým časem, viz (3.54). Transformace řešení je založena na perturbační metodě (4.3) s tokem (4.5).
$(P)ATMOVE - WA$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použito variačního přístupu s umělým časem, viz (3.54). Transformace řešení používá vážený průměr (4.10).
$(P)GSMOVE - PM$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použito variačního přístupu s Gaussovým-Seidelovým iteračním schématem, viz (3.58). Transformace řešení je založena na perturbační metodě (4.3) s tokem (4.5).
$(P)GSMOVE - WA$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použito variačního přístupu s Gaussovým-Seidelovým iteračním schématem, viz (3.58). Transformace řešení používá vážený průměr (4.10).
$(P)JACVQPMM - PM$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použita metoda minimalizace kvality vrcholu sítě, algoritmus 3.9, s Jacobiovým iteračním schématem (3.144). Transformace řešení je založena na perturbační metodě (4.3) s tokem (4.5).
$(P)JACVQPMM - WA$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použita metoda minimalizace kvality vrcholu sítě, algoritmus 3.9, s Jacobiovým iteračním schématem (3.144). Transformace řešení používá vážený průměr (4.10).
$(P)GSVQPMM - PM$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použita metoda minimalizace kvality vrcholu sítě, algoritmus 3.9, s Gaussovým-Seidelovým iteračním schématem (3.145). Transformace řešení je založena na perturbační metodě (4.3) s tokem (4.5).
$(P)GSVQPMM - WA$	Algoritmus 4.1(resp. 4.2). Pro adaptaci sítě je použita metoda minimalizace kvality vrcholu sítě, algoritmus 3.9, s Gaussovým-Seidelovým iteračním schématem (3.145). Transformace řešení používá vážený průměr (4.10).

Tabulka 4.1: Prefix  $(P)$  v označení metod značí použití algoritmu s predikcí, tedy algoritmu 4.2.

ve vztahu (3.75) pro výpočet ukazatele chyby L2E je použito řešení získané po-

mocí *metody charakteristik*. Integrál ve vztahu (3.75) je určen numericky pomocí čtyřbodové Gaussovy kvadratury na každém z objemů [35].

Jako numerický řešič jsme použili metodu konečných objemů s 1D MUSCL Lax-Friedrichsovým numerickým tokem, viz 2.6.2, s  $CFL$  číslem  $CFL = 0.3$ . Počet objemů sítě  $J = 30$  a v čase  $t = 0$  mají všechny objemy stejnou velikost. Adaptace sítě byla prováděna v každém kroku algoritmu 4.1 a 4.2, tedy parametr  $s = 1$ .

Pro adaptivní metody využívající variačního principu byla použita Winslowova monitorovací funkce (3.44) s parametrem  $\alpha = 0.2^9$ . U variačních metod s umělym časem bylo zvoleno  $CFL$  číslo pro podmínku stability (3.56)  $CFL = 0.5$ . Počet iterací pro úpravu sítě<sup>10</sup>  $MAXITER_{var}$  u metod založených na variačním principu byl zvolen  $MAXITER_{var} = 5$ . Na monitorovací funkci  $G$  bylo provedeno zhlazení (3.55).

Pro metody založené na minimalizaci parametru kvality vrcholu byly použity parametry ovlivňující adaptivitu  $\varepsilon_1 = 20$ ,  $p = 100$ , viz ((3.108) a (3.107)) a počet iterací pro úpravu sítě  $MAXITER_{vqpm}$  = 5. Na aproximaci Hessovy matice  $M$  bylo použito zhlazení (3.132) a reduční koeficient  $\gamma_r$  pro korekci (3.146) je volen  $\gamma_r = 0.99$ .

Metoda	$ITER$	$L2E$	$T$ [s]
<i>ATMOVE – PM</i>	20	0.002388	4.02
<i>ATMOVE – WA</i>	20	0.002627	1.32
<i>GSMOVE – PM</i>	20	0.001567	4.01
<i>GSMOVE – WA*</i>	20	0.001930	1.24
<i>JACVQPMM – PM</i>	23	0.001394	6.76
<i>JACVQPMM – WA*</i>	21	0.002072	2.27
<i>GSVQPMM – PM</i>	23	0.002004	7.06
<i>GSVQPMM – WA*</i>	21	0.001860	2.18

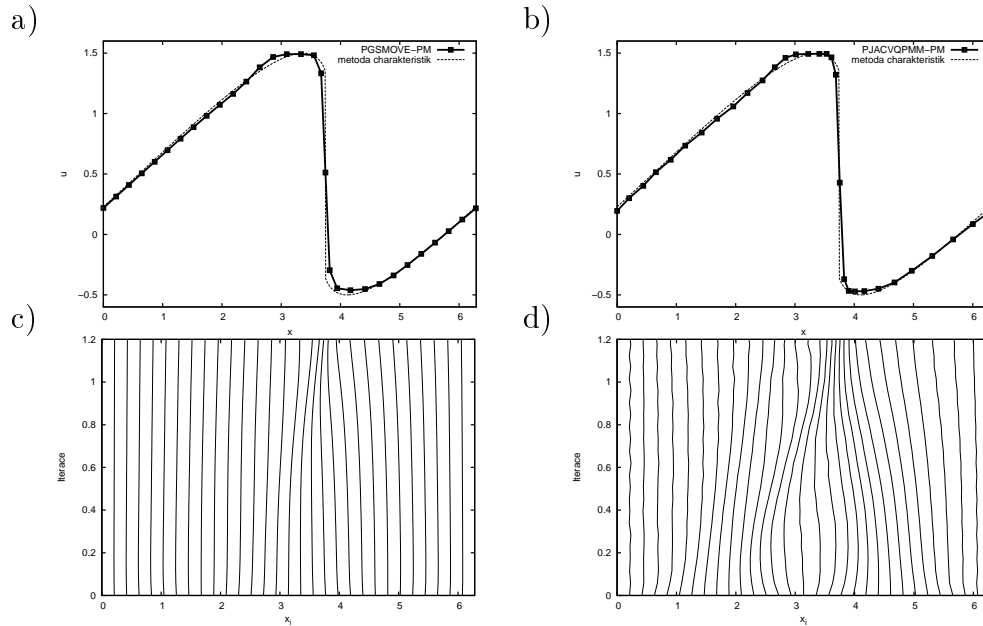
Tabulka 4.2: Sledované veličiny pro Burgersovu rovnici, metody bez predikce.

Metoda	$ITER$	$L2E$	$T$ [s]
<i>PATMOVE – PM</i>	20	0.002259	6.74
<i>PATMOVE – WA</i>	20	0.002481	1.73
<i>PGSMOVE – PM</i>	20	0.000659	6.75
<i>PGSMOVE – WA</i>	20	0.001742	1.65
<i>PJACVQPMM – PM</i>	22	0.001326	9.73
<i>PJACVQPMM – WA*</i>	21	0.001796	2.69
<i>PGSVQPMM – PM</i>	22	0.001832	9.69
<i>PGSVQPMM – WA</i>	21	0.001725	2.64

Tabulka 4.3: Sledované veličiny pro Burgersovu rovnici, metody s predikcí.

<sup>9</sup>Tato volba parametrů byla volena na základě článku [40].

<sup>10</sup>Protože v práci používáme značení  $MAXITER$  pro počet iterací proběhlých nad celou sítí jak u variačních metod, tak i u metod založených na minimalizaci parametru kvality vrcholu, a to, jak v 1D, tak i ve 2D, použijeme zde dolní index pro rozlišení metod. Tedy  $MAXITER_{var}$  je  $MAXITER$  pro variační metody a  $MAXITER_{vqpm}$  je  $MAXITER$  pro metody založené na minimalizaci parametru kvality vrcholu.



Obrázek 4.4: Burgersova rovnice. Nahoře (a),(b) je porovnání numerického řešení získaného pomocí metod *PGSMOVE – PM* (a) a *PJACVQPMM – PM* (b) s řešením obdržným metodou charakteristik. Dole (c),(d) jsou zachyceny trajektorie jednotlivých uzlů sítě při užití metod *PGSMOVE – PM* (c) a *PJACVQPMM – PM* (d).

Sledované veličiny pro fyzikální čas  $t = 1.2$ , tedy pro čas, kdy už došlo k vývoji postupující nespojitosti, jsou v tabulkách 4.2 a 4.3. Poznamenejme, že pro výpočet bez adaptivity byly sledované parametry následující:  $ITER = 21$ ,  $L2E = 0.004341$  a  $T = 0.48$  s. Pomocí „\*“ jsou v tabulkách označeny ty metody, při kterých došlo v průběhu experimentu k drobné oscilaci řešení v místě nespojitosti. Na obrázku 4.4 je zachyceno spočtené řešení pro vybrané metody a pohyb uzlů sítě v průběhu času.

Ze získaných hodnot je vidět, že při použití všech metod došlo k zhruba dvojnásobnému zvýšení přesnosti výpočtu, než bez použití adaptivity. Z hlediska přesnosti výpočtu vycházejí o něco lépe metody založené na minimalizaci parametru kvality vrcholu, důvodem je to, že v místě nespojitosti dochází k většímu zjemnění sítě. Toto ale vede k tomu, že díky CFL podmínce musí tyto metody udělat více časových kroků. Dále je vidět, že přesněji vycházejí transformace řešení, kde je použita perturbační metoda, drobné oscilace na nespojitosti vznikly pouze u přepočtu řešení, kde se používá vážený průměr (4.10).

Z hlediska časové náročnosti je vidět, že použití adaptivity vede k razantnímu zvětšení časové náročnosti výpočtu. Časově méně náročné se jeví použití metod založených na variačním principu<sup>11</sup>, důvodem je zřejmě opakovaná konstrukce Hessových matic a jejich aproximací v každém časovém kroku. Dále se ukazuje, že použití transformací řešení založených na váženém průměru je výrazně rychlejší než použití transformací řešení založených na perturbační metodě. Použití námi navrženého postupu s predikcí vedlo ke zvýšení přesnosti výpočtu v průměru o 14%, ale též i k jeho zpomalení v průměru o 40%, vůči algoritmu bez predikce.

<sup>11</sup>Připomeňme si, že v provedených experimentech pro předepsané funkce tomu bylo obráceně.

Při použití predikce též došlo k poklesu počtu výskytu drobných oscilací řešení v místě nespojitosti.

## Sodova trubice

V této části naší práce se zaměříme na otestování vlivu použitých řešičů na výslednou adaptivitu. Pomocí metody konečných objemů budeme numericky řešit Eulerovy rovnice (1.17) v 1D. Použité numerické toky jsou Roeův, Lax-Friedrichsův a Lax-Wendroffův numerický tok, které jsou uvedené v kapitole 2.6. Jako adaptivní algoritmy použijeme výše uvedené algoritmy, tabulka 4.1, ve verzi bez predikce. Jako testovací úloha nám bude sloužit tzv. *Sodova trubice*<sup>12</sup> [13, 28]. Jedná se o úlohu, jejíž fyzikální interpretace je taková, že velmi dlouhá tenká trubice naplněná plynem je rozdělena membránou na dvě části, kde v každé části je jiná hustota a tlak plynu. V čase  $t = 0$  pak dojde k odstranění této membrány.

Jako oblast  $\Omega$ , na které řešíme danou úlohu, uvažujme interval  $[-3, 3]$ . Počáteční podmínka je:

$$\mathbf{w}^0(x) = \begin{cases} (1, 0, 5)^T, & x < 0, \\ (1.101463, 0, 2.5)^T, & x > 0, \end{cases} \quad (4.17)$$

a okrajová podmínka je zvolena tak, že levý okraj oblasti je vstup a pravý výstup, viz [13, str. 203]. Fyzikální čas, do kterého je úloha řešena volíme  $t = 1$ .

Námi sledované parametry jsou počet proběhlých iterací algoritmu *ITER*, doba potřebná pro výpočet  $T$ , velikost největšího  $h_{max}$ , resp. nejmenšího  $h_{min}$ , objemu a průměrná velikost  $h_{avg}$  objemu. Dále budeme určovat chybu<sup>13</sup>, které jsme se dopustili pro složku řešení  $\mathbf{w}$ , jež odpovídá hustotě  $\rho$ , v časovém kroku  $t_k$ :

$$L2E_\rho = \sqrt{\sum_{i=1}^J \int_{x_i}^{x_{i+1}} (\rho(x, t_k) - \rho_i^k)^2 dx}, \quad (4.18)$$

kde integrál na pravé straně je určen numericky, pomocí čtyřbodové Gaussovy kvadratury na každém z objemů [35], a jako přesné řešení použijeme řešení uvedené v [13, str. 137].

Pro všechny řešiče jsme použili  $CFL = 0.3$ . Protože v případě použití Lax-Wendroffova numerického toku docházelo k oscilacím, tak jsme navíc ještě otestovali následující modifikaci Lax-Wendroffova numerického toku. Pomocí vztahů (4.6) jsme spočetli lineární rekonstrukce  $\mathbf{w}_i^{+,k}$ ,  $\mathbf{w}_i^{-,k}$  a  $\mathbf{w}_{i+1}^{+,k}$ ,  $\mathbf{w}_{i+1}^{-,k}$  v uzlových bodech sítě  $x_i$  a  $x_{i+1}$  a tyto hodnoty použili v numerickém toku (2.77) Lax-Wendroffova schématu, viz kapitola 2.6.3. Poznamenejme, že při konstrukci hodnot  $\mathbf{w}_i^{\pm,k}$  je použit limiter (4.8), z tohoto důvodu budeme toto schéma nazývat Lax-Wendroffův numerický tok s limiterem. Tento postup vedl ke zmenšení<sup>14</sup> oscilací v okolí nespojitosti, viz obrázek 4.5 vlevo. Adaptace sítě byla prováděna

<sup>12</sup>shock tube problem

<sup>13</sup>Protože budeme provádět výpočty na sítích s různým počtem objemů, bude přehlednější sledovat celkovou chybu, které se dopustíme, nikoliv chybu vztahenou na jeden objem, viz definice ukazatele L2E (3.75).

<sup>14</sup>Cílem naší práce je prověřit navržené metody adaptace, nikoliv testování vhodných numerických toků pro tento problém, z toho důvodu se spokojíme s výše uvedenými numerickými toky.



v každém kroku algoritmu 4.1, tedy parametr  $s = 1$ . Použité sítě obsahovaly  $J = 20, 40, 80, 160, 320$  objemů a velikosti všech objemů na příslušné síti byly v časovém okamžiku  $t = 0$  stejné.

Pro adaptivní metody využívající variačního principu byla použita Winslowova monitorovací funkce (3.44) s parametrem  $\alpha = 1$ , pro  $J = 20, 40, 80$  objemů a  $\alpha = 0.1$  pro  $J = 160, 320$  objemů. U variačních metod s umělým časem bylo pro podmínku stability (3.56) zvoleno  $CFL$  číslo,  $CFL = 0.5$ . Počet iterací pro úpravu sítě,  $MAXITER_{var}$  byl zvolen,  $MAXITER_{var} = 1$ . Na monitorovací funkci  $G$  bylo provedeno zhlazení (3.55).

Pro metody založené na minimalizaci parametru kvality vrcholu byly použity následující parametry  $\varepsilon_1$  a  $p$ , viz ((3.108) a (3.107)),  $\varepsilon_1 = 100$ ,  $p = 2$  pro  $J = 20, 40, 80$  objemů a  $\varepsilon_1 = 10$ ,  $p = 2$  pro  $J = 160, 320$  objemů. Počet iterací pro úpravu sítě  $MAXITER_{vqpm} = 1$ . Na aproximaci Hessovy matice  $M$  bylo použito zhlazení (3.132) a reduční koeficient  $\gamma_r$  pro korekci (3.146) je volen  $\gamma_r = 0.99$ .

Důvodem ke změně parametrů adaptivity je v obou případech ten fakt, že v případě použití Lax-Wendroffova či Roeova numerického toku a většího počtu objemů, byly některé objemy příliš malé a díky  $CFL$  podmínce trval výpočet neúměrně dlouho<sup>15</sup>, řádově hodiny.

Získané výsledky jsou zachyceny v tabulkách 4.4, 4.5, 4.6 a 4.7, kde vždy prvních pět řádků udává výsledky získané za použití příslušného numerického toku bez adaptivity. Pro přehlednější vzájemné porovnání výsledků jsou v příloho­vé části získané hodnoty zaneseny do grafů 4.13 až 4.17. Zde je vidět, že v případě Lax-Friedrichsova numerického toku nevede použití adaptivity k přesnějším výsledkům (z hlediska ukazatele  $L2E_\rho$ ). V případě Roeova numerického toku nevede vždy k přesnějšímu výpočtu. V případě použití Lax-Wendroffova numerického toku vede ke zvýšení přesnosti výpočtu. V případě použití Lax-Wendroffova numerického toku s limiterem dochází ke zvýšení přesnosti výpočtu ve všech případech jen u metod  $GSMOVE - PM$  a  $GSVQPMM - PM$ , což je námi navržená metoda, viz tabulka 4.1. Ze spočtených výsledků vyplývá, že použité numerické schéma, v algoritmu 4.1, musí být schopno dostatečně ostře zachytit nespojitost, což např. v případě Lax-Friedrichsova numerického toku není díky difuzním vlastnostem tohoto numerického toku splněno.

Dále je z výsledků patrné, že dochází ke změně chování metod při přechodu od sítě s  $J = 80$  objemů k sítím s  $J = 160, 320$  objemů, tedy při změně parametrů ovlivňujících adaptivitu. Ze získaných výsledků lze nahlédnout, že přesnějších výsledků dosahují metody, které používají pro přepočítání řešení vztahu (4.3), tedy metody používající perturbační metodu<sup>16</sup>. Na obrázku 4.5 vpravo je zaneseno porovnání průměrných časů potřebných k jednomu adaptačnímu kroku, vztažených na jeden objem sítě, pro všechny použité metody. Zde je vidět, že v tomto případě je čas potřebný pro jeden adaptační krok nižší o cca 60% u metod vycházejících z variačního principu. Pro Lax-Wendroffův numerický tok, či Lax-Wendroffův numerický tok s limiterem, tedy v případech, kdy se uplatnil vliv adaptivity, je celková doba běhu výpočtu většinou nižší<sup>17</sup> při použití našich metod, v některých

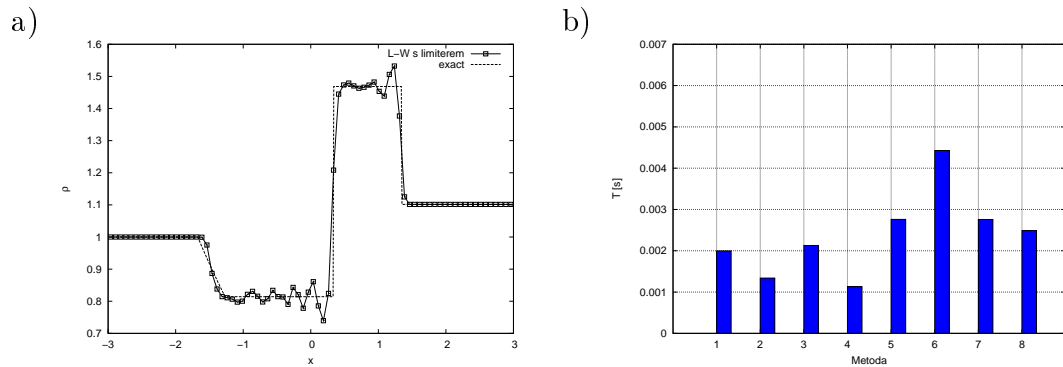
<sup>15</sup>Z části je to způsobeno tím, že použitý implementační jazyk Octave je poměrně vysokoúrovňový interpretovaný jazyk, a díky tomu je výsledný program mnohem pomalejší, než např. program napsaný v jazyce C, či Fortranu.

<sup>16</sup>Tedy metody mající sufix  $PM$ .

<sup>17</sup>Námi navržené metody vykonávají menší počet výpočetních kroků algoritmu 4.1, proto jsou v konečném důsledku rychlejší.

případech i o více než 100%. Při porovnání dosažených výsledků z hlediska přesnosti, v případě použití Lax-Wendroffova numerického toku, či v Lax-Wendroffova numerického toku s limiterem, jsou naše navržené metody v průměru přesnější.

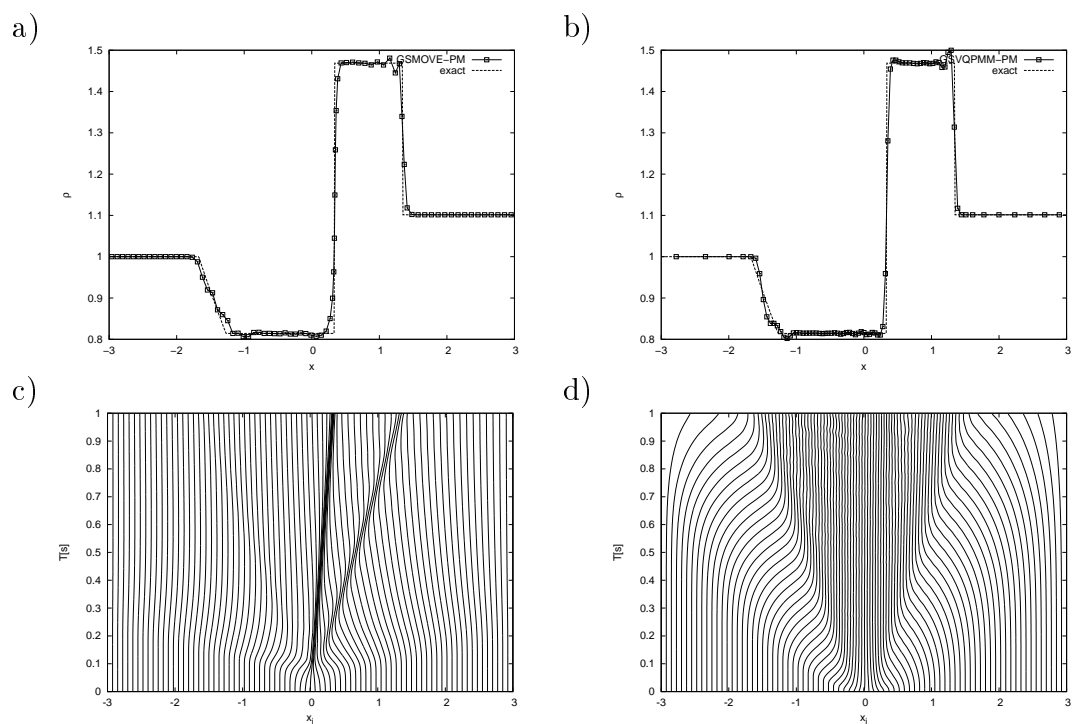
Porovnání řešení získaných za použití Lax-Wendroffova numerického toku s limiterem v kombinaci s adaptivitou je zaneseno pro  $J = 80$  objemů na obrázku 4.6. Při srovnání s řešením získaným bez adaptivity, viz obrázek 4.5, je patrné, že použití adaptivity zmenšilo oscilace poblíž nespojitosti. Ze srovnání trajektorií jednotlivých uzlů sítě<sup>18</sup> je vidět, že metody využívající variačního principu ostřeji zachycují nespojitost, zatímco metody založené na minimalizaci parametru kvality vrcholu, uzly sítě rozmístí po celé oblasti<sup>19</sup>.



Obrázek 4.5: (a) Spočtené řešení pro Sodovu trubici, použit Lax-Wendroffův numerický tok s limiterem, v čase  $t = 1$  pro  $J = 80$  objemů. (b) Porovnání průměrných časů potřebných pro jeden adaptační krok pro testované metody. Metody jsou očíslovány dle pořadí v tabulce 4.1

<sup>18</sup>Při takto zvolených parametrech  $\alpha = 1$  pro variační metody a  $\varepsilon_1 = 100$ ,  $p = 2$  pro metody založené na minimalizaci parametru kvality vrcholu.

<sup>19</sup>Poznamenejme, že přesnější řešení ve smyslu  $L2E_\rho$  je dosaženo metodou *GSVQPMM - PM*, tedy na obrázku 4.6 vpravo



Obrázek 4.6: Porovnání spočtených řešení pro Sodovu trubici, Lax-Wendroffův numerický tok s limiterem,  $J = 80$ . Nahoře (a),(b) je porovnání numerického řešení získaného pomocí metod  $GSMOVE-PM$  (a) a  $GSVQPMM-PM$  (b) s přesným řešením v čase  $t = 1$ . Dole (c),(d) jsou zachyceny trajektorie jednotlivých uzlů sítě při užití metod  $GSMOVE-PM$  (c) a  $GSVQPMM-PM$  (d).

Metoda	J	$h_{min}$	$h_{max}$	$h_{avg}$	$ITER$	$L2E_\rho$	$T$ [s]
Lax-Friedrichs	20	0.3	0.3	0.3	20	0.453762	0.43
	40	0.15	0.15	0.15	39	0.389461	1.57
	80	0.075	0.075	0.075	80	0.321993	6.34
	160	0.0375	0.0375	0.0375	164	0.259846	26.06
	320	0.01875	0.01875	0.01875	335	0.211660	108.16
ATMOVE-PM	20	0.299418	0.3003	0.3	20	0.453695	1.36
	40	0.148876	0.150639	0.15	40	0.390128	5.07
	80	0.073239	0.076064	0.075	82	0.324222	20.16
	160	0.037273	0.03763	0.0375	166	0.260768	80.86
	320	0.018523	0.01886	0.01875	344	0.212414	337.80
ATMOVE-MM	20	0.299418	0.3003	0.3	20	0.453695	0.97
	40	0.148876	0.150639	0.15	40	0.390128	3.62
	80	0.07324	0.076064	0.075	82	0.324345	14.74
	160	0.037273	0.03763	0.0375	166	0.260768	61.56
	320	0.018523	0.01886	0.01875	344	0.212414	267.39
GSMOVE-PM	20	0.299195	0.300357	0.3	20	0.453563	1.35
	40	0.1484	0.150633	0.15	40	0.389923	5.02
	80	0.072531	0.076203	0.075	83	0.324592	20.26
	160	0.037183	0.037677	0.0375	166	0.260461	80.41
	320	0.018419	0.018923	0.01875	348	0.213166	338.96
GSMOVE-MM	20	0.299195	0.300357	0.3	20	0.453563	0.96
	40	0.1484	0.150633	0.15	40	0.389923	3.29
	80	0.072535	0.076202	0.075	83	0.324715	13.30
	160	0.037183	0.037677	0.0375	166	0.260461	55.59
	320	0.01842	0.018923	0.01875	348	0.213166	256.60
JACVQPMM-PM	20	0.182279	0.621596	0.3	33	0.433843	2.67
	40	0.091386	0.346239	0.15	71	0.401945	10.72
	80	0.040897	0.230057	0.075	162	0.355190	47.70
	160	0.02804	0.052211	0.0375	210	0.262906	123.10
	320	0.014698	0.024664	0.01875	398	0.212414	469.15
JACVQPMM-MM	20	0.175005	0.608151	0.3	33	0.432874	2.31
	40	0.091758	0.439977	0.15	70	0.395676	13.50
	80	0.037101	0.286247	0.075	173	0.354964	102.22
	160	0.028239	0.053623	0.0375	210	0.264423	256.34
	320	0.014677	0.026843	0.01875	397	0.213915	1447.36
GSVQPMM-PM	20	0.153983	0.693338	0.3	39	0.435798	3.12
	40	0.076242	0.257803	0.15	78	0.369107	11.70
	80	0.04196	0.216756	0.075	153	0.292027	44.96
	160	0.024012	0.09389	0.0375	248	0.261380	144.68
	320	0.013593	0.028374	0.01875	447	0.210902	525.14
GSVQPMM-MM	20	0.157143	0.760718	0.3	39	0.437607	2.35
	40	0.084043	0.259568	0.15	77	0.372182	9.80
	80	0.042456	0.218299	0.075	150	0.295431	42.72
	160	0.025458	0.095637	0.0375	245	0.263818	197.69
	320	0.013545	0.029156	0.01875	448	0.213166	979.27

Tabulka 4.4: Sledované veličiny pro Sodovu trubici, Lax-Friedrichsův numerický tok,  $t = 1$ .

Metoda	J	$h_{min}$	$h_{max}$	$h_{avg}$	$ITER$	$L2E_\rho$	$T$ [s]
Roe	20	0.3	0.3	0.3	22	0.238327	1.63
	40	0.15	0.15	0.15	44	0.188679	6.35
	80	0.075	0.075	0.075	87	0.156204	24.99
	160	0.0375	0.0375	0.0375	173	0.124579	98.92
	320	0.01875	0.01875	0.01875	346	0.102761	397.45
ATMOVE-PM	20	0.265492	0.312947	0.3	23	0.234179	2.76
	40	0.108131	0.170708	0.15	58	0.185903	13.29
	80	0.038579	0.096621	0.075	173	0.142828	77.99
	160	0.030367	0.040852	0.0375	243	0.122637	218.61
	320	0.012718	0.022548	0.01875	680	0.096333	1221.77
ATMOVE-MM	20	0.265964	0.312889	0.3	23	0.235669	2.17
	40	0.11164	0.169682	0.15	57	0.190787	10.52
	80	0.044454	0.094103	0.075	160	0.151789	60.44
	160	0.031774	0.040525	0.0375	232	0.127121	181.59
	320	0.014444	0.021988	0.01875	582	0.101192	926.49
GSMOVE-PM	20	0.257588	0.308961	0.3	24	0.230738	2.87
	40	0.098525	0.159481	0.15	67	0.182866	15.38
	80	0.023846	0.085171	0.075	320	0.130843	142.54
	160	0.028352	0.039076	0.0375	285	0.12	252.40
	320	0.008439	0.020763	0.01875	1411	0.087635	2460.89
GSMOVE-MM	20	0.258895	0.308722	0.3	24	0.233066	2.24
	40	0.104767	0.158811	0.15	65	0.190997	11.99
	80	0.032992	0.083972	0.075	265	0.148323	98.85
	160	0.030351	0.038921	0.0375	260	0.127121	195.85
	320	0.012097	0.020496	0.01875	979	0.099599	1486.70
JACVQPM-PM	20	0.188502	0.533713	0.3	37	0.215127	4.92
	40	0.073327	0.460453	0.15	87	0.161369	22.18
	80	0.020958	0.436104	0.075	275	0.130537	138.22
	160	0.016571	0.095362	0.0375	299	0.117983	298.86
	320	0.007336	0.054682	0.01875	678	0.096332	1356.85
JACVQPM-MM	20	0.186513	0.527291	0.3	35	0.228779	3.98
	40	0.080578	0.433596	0.15	83	0.181659	23.01
	80	0.032347	0.399128	0.075	194	0.158996	150.50
	160	0.018268	0.094932	0.0375	279	0.125219	335.11
	320	0.00822	0.055835	0.01875	637	0.101192	1891.04
GSVQPM-PM	20	0.198987	0.497277	0.3	41	0.225743	5.44
	40	0.075925	0.409302	0.15	108	0.163707	27.52
	80	0.022593	0.39893	0.075	274	0.129614	137.69
	160	0.014517	0.163324	0.0375	349	0.111713	349.48
	320	0.006622	0.059258	0.01875	771	0.091214	1518.34
GSVQPM-MM	20	0.201987	0.475502	0.3	39	0.243392	4.18
	40	0.0798	0.380963	0.15	98	0.183738	20.68
	80	0.022673	0.356727	0.075	254	0.146424	107.92
	160	0.016064	0.165201	0.0375	323	0.118659	308.44
	320	0.007212	0.066531	0.01875	721	0.096332	1463.32

Tabulka 4.5: Sledované veličiny pro Sodovu trubici, Roeův numerický tok,  $t = 1$ .

Metoda	J	$h_{min}$	$h_{max}$	$h_{avg}$	ITER	$L2E_\rho$	T [s]
Lax-Wendroff	20	0.3	0.3	0.3	24	0.252586	1.01
	40	0.15	0.15	0.15	48	0.199298	3.86
	80	0.075	0.075	0.075	95	0.158240	15.18
	160	0.0375	0.0375	0.0375	191	0.121983	61.05
	320	0.01875	0.01875	0.01875	381	0.097979	245.36
ATMOVE-PM	20	0.226092	0.325351	0.3	27	0.215824	1.03
	40	0.077589	0.182529	0.15	76	0.165770	12.55
	80	0.019609	0.122227	0.075	286	0.125538	92.87
	160	0.016904	0.049369	0.0375	404	0.116619	260.35
	320	0.010922	0.027181	0.01875	818	0.071554	1054.11
ATMOVE-MM	20	0.227756	0.325381	0.3	27	0.211471	1.69
	40	0.085274	0.181028	0.15	72	0.161740	8.77
	80	0.030467	0.110975	0.075	222	0.117983	57.56
	160	0.024634	0.044551	0.0375	312	0.098792	172.40
	320	0.003701	0.03435	0.01875	1795	0.083904	2337.91
GSMOVE-PM	20	0.205389	0.322112	0.3	29	0.196723	2.54
	40	0.059278	0.171542	0.15	105	0.148189	17.38
	80	0.006869	0.099173	0.075	797	0.089888	260.60
	160	0.012498	0.043468	0.0375	583	0.093808	378.23
	320	0.000208	0.023043	0.01875	22198	0.04	31226.28
GSMOVE-MM	20	0.208413	0.319576	0.3	28	0.198191	1.69
	40	0.075286	0.167344	0.15	93	0.153362	11.25
	80	0.019663	0.091315	0.075	471	0.106207	117.84
	160	0.023349	0.04132	0.0375	378	0.092086	193.82
	320	0.007832	0.023527	0.01875	1723	0.066932	1744.19
JACVQPM-PM	20	0.183752	0.644194	0.3	43	0.200349	4.35
	40	0.070681	0.598469	0.15	111	0.148862	21.36
	80	0.0287	0.627552	0.075	281	0.113490	105.55
	160	0.026779	0.110179	0.0375	268	0.108074	193.95
	320	0.011043	0.062541	0.01875	535	0.087635	793.35
JACVQPM-MM	20	0.186418	0.617412	0.3	41	0.213635	3.36
	40	0.076613	0.556608	0.15	100	0.162972	15.53
	80	0.031312	0.524091	0.075	240	0.123612	149.57
	160	0.02707	0.108649	0.0375	262	0.105830	259.22
	320	0.009124	0.059936	0.01875	546	0.085790	1562.26
GSVQPM-PM	20	0.202349	0.619444	0.3	45	0.179888	4.55
	40	0.074284	0.581476	0.15	125	0.133416	24.08
	80	0.025817	0.602178	0.075	310	0.102371	117.30
	160	0.022586	0.139844	0.0375	310	0.101192	233.72
	320	0.008724	0.074644	0.01875	641	0.083904	972.68
GSVQPM-MM	20	0.199091	0.546166	0.3	42	0.190052	3.32
	40	0.07734	0.49012	0.15	112	0.146560	17.04
	80	0.027989	0.487195	0.075	275	0.116275	84.23
	160	0.022372	0.144782	0.0375	300	0.097159	224.26
	320	0.007892	0.076701	0.01875	643	0.077974	1049.72

Tabulka 4.6: Sledované veličiny pro Sodovu trubici, Lax-Wendroffův numerický tok,  $t = 1$ .

Metoda	J	$h_{min}$	$h_{max}$	$h_{avg}$	ITER	$L2E_\rho$	$T$ [s]
Lax-Wendroff-lim	20	0.3	0.3	0.3	23	0.197119	2.06
	40	0.15	0.15	0.15	45	0.119465	7.80
	80	0.075	0.075	0.075	90	0.104511	30.68
	160	0.0375	0.0375	0.0375	180	0.056937	121.97
	320	0.01875	0.01875	0.01875	358	0.044654	475.90
ATMOVE-PM	20	0.243171	0.317754	0.3	26	0.178513	3.51
	40	0.079226	0.187756	0.15	72	0.116817	18.54
	80	0.018143	0.124814	0.075	283	0.078901	143.64
	160	0.015452	0.050082	0.0375	406	0.062708	411.44
	320	0.003993	0.034300	0.018750	1675	0.041512	3310.63
ATMOVE-MM	20	0.245045	0.316940	0.3	26	0.176715	2.85
	40	0.087187	0.184198	0.15	69	0.132927	15.04
	80	0.032501	0.111198	0.075	209	0.099340	93.80
	160	0.024728	0.045867	0.0375	303	0.080667	283.49
	320	0.011444	0.027642	0.01875	762	0.066352	1572.81
GSMOVE-PM	20	0.232470	0.321771	0.3	27	0.172709	3.63
	40	0.060848	0.173761	0.15	96	0.103122	24.62
	80	0.006575	0.098831	0.075	775	0.060808	394.09
	160	0.010459	0.042146	0.0375	614	0.053559	619.08
	320	0.000497	0.023974	0.018750	11831	0.029755	24293.16
GSMOVE-MM	20	0.236722	0.320276	0.3	27	0.171703	2.91
	40	0.073575	0.170169	0.15	85	0.124474	18.55
	80	0.020255	0.092603	0.075	422	0.101586	182.14
	160	0.023154	0.041633	0.0375	362	0.085217	325.76
	320	0.007799	0.024503	0.018750	1606	0.065379	2876.45
JACVQPMM-PM	20	0.165872	0.700471	0.3	47	0.144090	6.95
	40	0.056634	0.693609	0.15	140	0.097405	39.69
	80	0.045232	0.207556	0.075	140	0.071057	78.00
	160	0.024045	0.124992	0.0375	274	0.059346	304.83
	320	0.013157	0.072450	0.018750	526	0.048012	1176.55
JACVQPMM-MM	20	0.176579	0.653232	0.3	41	0.176302	5.34
	40	0.068100	0.600064	0.15	109	0.122266	32.29
	80	0.044970	0.205775	0.075	138	0.068624	75.09
	160	0.023352	0.120571	0.0375	270	0.060967	331.17
	320	0.012566	0.069844	0.01875	519	0.045739	1668.36
GSVQPMM-PM	20	0.193438	0.623881	0.3	53	0.153405	7.87
	40	0.065367	0.700356	0.15	152	0.095727	43.32
	80	0.036950	0.444010	0.075	166	0.061656	92.36
	160	0.018361	0.164538	0.0375	331	0.039426	369.36
	320	0.011395	0.097696	0.01875	612	0.042631	1331.44
GSVQPMM-MM	20	0.200065	0.550466	0.3	45	0.174558	5.67
	40	0.065781	0.527804	0.15	125	0.136795	30.49
	80	0.034054	0.473261	0.075	164	0.087704	84.27
	160	0.018628	0.160454	0.0375	324	0.061407	361.80
	320	0.011175	0.095626	0.01875	603	0.048324	1416.48

Tabulka 4.7: Sledované veličiny pro Sodovu trubici, Lax-Wendroffův numerický tok s limiterem,  $t = 1$ .

## Modifikovaný problém Sodovy trubice

V dalším testu uvažujme modifikovaný problém Sodovy trubice. Jako oblast  $\Omega$ , na které řešíme danou úlohu, volíme interval  $[0, 5]$ . Počáteční podmínka je :

$$\mathbf{w}^0(x) = \begin{cases} (10, 0, 250)^T, & x \leq 2, \\ (1, 0, 2.5)^T, & x > 2, \end{cases} \quad (4.19)$$

a okrajová podmínka je zvolena tak, že levý okraj oblasti je vstup a pravý výstup, fyzikální čas, do kterého je úloha řešena, volíme  $t = 0.4$ . Jako řešič použijeme metodu konečných objemů s Lax-Wendroffovým numerickým tokem s limiterem, který dosáhl pro Sodovu trubici nejlepších výsledků, viz předchozí numerický experiment, použijeme *CFL* číslo  $CFL = 0.3$ .

Protože jsme se v předchozím experimentu setkali se závislostí adaptivního procesu na volbě použitých parametrů<sup>20</sup>, vyberme si dva zástupce testovaných metod a prověříme vliv volby použitých parametrů na adaptivní proces. Jako zástupce variačních metod volíme metodu *GSMOVE – PM* a jako zástupce námi navržených metod, založených na minimalizaci parametru kvality vrcholu, volíme metodu *GSVQPMM – PM*, viz tabulka 4.1, které dosáhly nejlepších výsledků z hlediska  $L2E_\rho$ , viz (4.18).

Sledované veličiny jsou počet proběhlých iterací algoritmu *ITER*, doba potřebná pro výpočet  $T$ , velikost největšího  $h_{max}$ , resp. nejmenšího  $h_{min}$ , objemu a průměrná velikost  $h_{avg}$  objemu. Chybu spočteného numerického řešení  $\mathbf{w}^k$  v čase  $t_k$  budeme určovat pomocí  $L2E_\rho$ , vztah (4.18). Postupujeme následovně, nejprve spočteme po částech konstantní řešení na síti s  $J = 1000$  objemů pomocí přesného Riemannova řešiče pro tento problém, síťové body této sítě značíme  $\tilde{x}_j$ ,  $j = 1, \dots, 1001$  a příslušnou hustotu na  $j$ -tém objemu v čase  $t_k$  označíme jako  $\tilde{\rho}_j^k$ . Pak pro  $x \in [\tilde{x}_j, \tilde{x}_{j+1}]$  použijeme při vyhodnocení vztahu (4.18) místo  $\rho(x, t_k)$  hodnotu  $\tilde{\rho}_j^k$ . Poznamenejme, že jako přesný Riemannův řešič byl použit řešič implementovaný v programu<sup>21</sup> [41].

Otestovali jsme výše uvedené metody na síti s  $J = 160$  objemů, velikosti všech objemů byly v čase  $t = 0$  stejné. Adaptace sítě byla prováděna v každém kroku algoritmu 4.1, tedy parametr  $s = 1$ . Pro metodu *GSMOVE – PM* jsme volili počet iterací pro úpravu sítě,  $MAXITER_{var} = 1$ . Parametr  $\alpha$  pro Winslowovu monitorovací funkci (3.44) nabýval hodnot uvedených v tabulce 4.8. Na monitorovací funkci  $G$  bylo provedeno zhlazení (3.55). Pro metodu *GSVQPMM – PM* byly postupně použity volby parametrů dané tabulkou 4.10. Počet iterací pro úpravu sítě  $MAXITER_{vqpm} = 1$ . Na aproximaci Hessovy matice  $\mathbb{M}$  bylo použito zhlazení (3.132) a reduční koeficient  $\gamma_r$  pro korekci (3.146) je volen  $\gamma_r = 0.99$ .

Ze spočtených hodnot uvedených v tabulkách 4.8 a 4.10 je vidět, že celý adaptační proces je závislý na použitých parametrech. Poznamenejme, že volba testovaných parametrů u obou metod byla provedena experimentálně tak, aby doba výpočtu byla pro většinu variant řádově stejná.

U metody *GSMOVE – PM*, viz tabulka 4.8, je vidět velká citlivost adaptačního procesu na uživatelem zvolený parametr  $\alpha$ . Výrazně vyšší doba běhu algoritmu pro volbu  $\alpha = 0.002$  je dána tím, že zhruba v čase  $T = 0.15$ , je velikost

<sup>20</sup>Při změně parametrů  $\alpha$  resp.  $\varepsilon_1$  a  $p$  pro variační metody, resp. pro metody založené na minimalizaci parametru kvality vrcholu.

<sup>21</sup>Ke stažení na: [http://cococubed.asu.edu/code\\_pages/exact\\_riemann.shtml](http://cococubed.asu.edu/code_pages/exact_riemann.shtml)



$\alpha$	$J$	$h_{min}$	$h_{max}$	$h_{avg}$	$ITER$	$L2E_\rho$	$T$ [s]
0.001	160	0.016134	0.039090	0.031250	584	0.594960	587.69
0.0015	160	0.011655	0.043231	0.031250	870	0.673742	872.98
0.002	160	0.020382	0.054772	0.031250	4893	0.715616	5039.86
0.002 (DS)	160	0.011842	0.042446	0.031250	880	0.682497	891.70

Tabulka 4.8: Sledované veličiny pro modifikovaný Sodův problém v  $t = 0.4$ , pro  $J = 160$  objemů, Lax-Wendroffův numerický tok s limiterem s použitím adaptivity, metoda *GSMOVE – PM*. Pomocí DS jsou označeny hodnoty získané pomocí dynamického škálování.

nejmenšího objemu sítě velmi malá, řádově  $10^{-4}$ , čímž dochází v důsledku splnění CFL podmínky ke zmenšení časového kroku a metoda musí vykonat mnohem větší počet kroků, než dospěje do koncového času. Tento jev je těžko předvídatelný a částečně zhoršuje použitelnost metody, viz obdržené výsledky pro předchozí úlohu v tabulkách 4.6 a 4.7 pro  $J = 320$  objemů u této metody.

Možným řešením je dynamicky škálovat parametr  $\alpha$  v monitorovací funkci (3.44) dle hodnot časového kroku  $\tau_k$ , viz (2.27), metody konečných objemů. Výsledky obdržené pomocí dynamického škálování jsou v tabulce 4.8 označeny DS. Dynamické škálování parametru  $\alpha$  je realizováno tak, že pokud časový krok  $\tau_k$  je v určitých předepsaných mezích, pak se upraví hodnota  $\alpha$  tak, že nová hodnota  $\tilde{\alpha} = c\alpha$ , kde  $c$  je předepsaná konstanta. Použili jsme volbu mezí a konstant  $c$  danou tabulkou 4.9. Z dosažených výsledků je vidět, že dynamické škálování parametru  $\alpha$  tento jev eliminuje.

meze časového kroku $\tau_k$	$c$
$\tau_k \geq 10^{-2}$	10
$\tau_k \in (10^{-2}, 10^{-3})$	1.1
$\tau_k \in (10^{-3}, 2 \cdot 10^{-4})$	1
$\tau_k \leq 2 \cdot 10^{-4}$	0.9

Tabulka 4.9: Volba konstanty  $c$  pro určení  $\alpha$  při dynamickém škálování.

Porovnáním s výsledky pro  $J = 320$  objemů, za použití stejného řešiče, ale bez použití adaptivity:  $ITER = 638$ ,  $L2E_\rho = 0.668326$  a  $T = 870.01$  s, je vidět, že pro první dvě volby parametru  $\alpha = 0.001$ ,  $\alpha = 0.0015$  a pro  $\alpha = 0.002$  při použití dynamického škálování jsou spočtené výsledky srovnatelné, ale při  $J = 160$  objemů.

U námi navržené metody *GSVQPMM – PM* vede zvyšování parametru  $p$ , při stejném parametru  $\varepsilon_1$ , k růstu doby potřebné pro celý výpočet, viz tabulka 4.10. Obdobně se metoda chová i při růstu parametru  $\varepsilon_1$ , při stejné hodnotě  $p$ . Tento nárůst doby potřebné pro výpočet je opět způsoben tím, že se v síti s růstem výše uvedených parametrů vyskytují menší a menší objemy, a tak, vzhledem k podmínce stability, dochází ke zmenšování časového kroku, a následně k nutnosti vykonat větší počet výpočetních kroků. Výše zmíněný jev skokového zpomalení výpočtu při malé změně parametrů metody, viz tabulka 4.8, nebyl u naší metody zaznamenán. Porovnáním s výsledky pro  $J = 320$  objemů, za použití stejného řešiče, ale bez použití adaptivity, viz výše, vidíme, že námi dosažené výsledky

$\varepsilon_1$	$p$	$J$	$h_{min}$	$h_{max}$	$h_{avg}$	$ITER$	$L2E_\rho$	$T$ [s]
10	1	160	0.017933	0.261659	0.03125	576	0.358931	635.49
10	2	160	0.017745	0.259817	0.03125	578	0.366693	639.35
10	5	160	0.017861	0.255138	0.03125	579	0.274215	639.38
10	10	160	0.017744	0.261199	0.03125	581	0.275069	641.80
10	20	160	0.017764	0.268416	0.03125	582	0.340231	643.47
10	50	160	0.013745	0.260825	0.03125	629	0.328261	697.53
10	100	160	0.011598	0.247890	0.03125	677	0.333312	749.86
20	1	160	0.014390	0.324429	0.03125	745	0.475071	799.62
20	2	160	0.014323	0.327197	0.03125	749	0.499391	819.74
20	5	160	0.014095	0.333358	0.03125	753	0.497429	824.04
20	10	160	0.013786	0.339112	0.03125	755	0.503116	827.67
20	20	160	0.013349	0.345264	0.03125	760	0.507412	835.61
20	50	160	0.012878	0.354638	0.03125	764	0.418721	839.09
20	100	160	0.011954	0.360737	0.03125	768	0.400549	832.22
30	1	160	0.012550	0.324391	0.03125	879	0.488508	968.63
30	2	160	0.012474	0.327922	0.03125	883	0.509361	977.17
30	5	160	0.012142	0.335662	0.03125	890	0.517467	982.61
30	10	160	0.011532	0.342694	0.03125	897	0.505701	990.99
30	20	160	0.011902	0.351123	0.03125	902	0.496453	996.96
30	50	160	0.011511	0.362301	0.03125	912	0.481553	1009.54
30	100	160	0.011055	0.369289	0.03125	924	0.474962	1027.22

Tabulka 4.10: Sledované veličiny pro modifikovaný Sodův problém v  $t = 0.4$ , pro  $J = 160$  objemů, Lax-Wendroffův numerický tok s limiterem s použitím adaptivity, metoda *GSVQPMM – PM*.

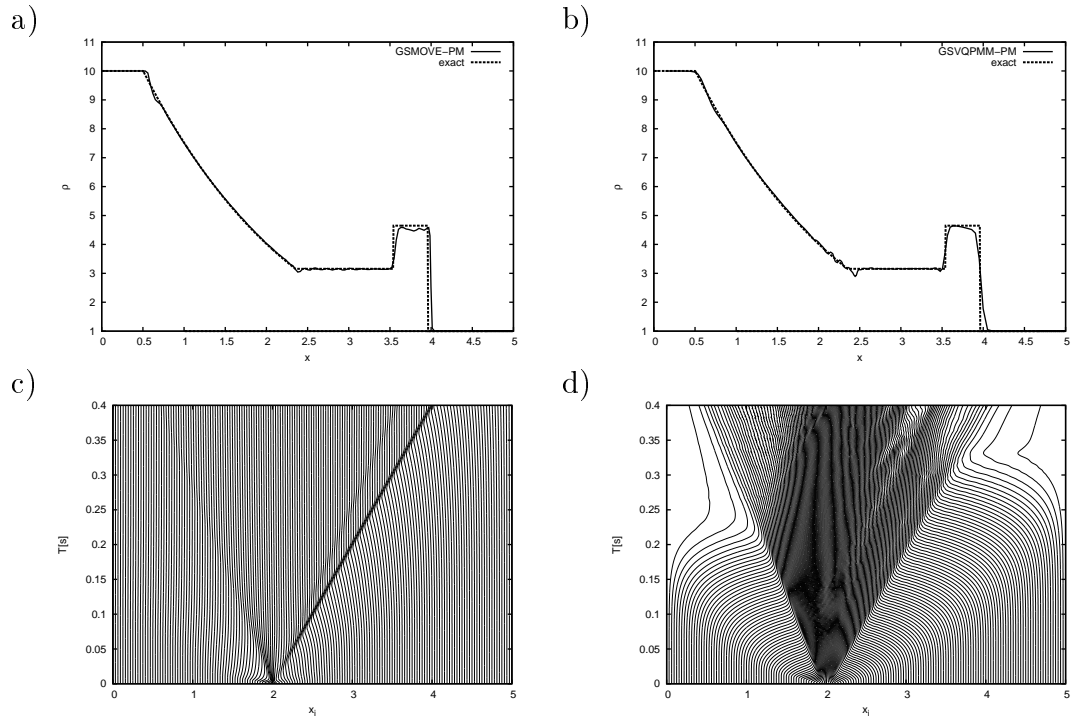
jsou při zhruba stejné době výpočtu  $T$ , za použití  $J = 160$  objemů z hlediska ukazatele  $L2E_\rho$ , přesnější.

Na obrázcích 4.7 je zaneseno porovnání spočtených výsledků s přesným řešením. Zde je patrné, že řešení získané pomocí adaptivních metod odpovídá přesnému řešení. Obdobně jako v předchozím numerickém testu vedlo použití adaptivity ke zmenšení oscilací, jež produkuje samotný řešič bez použití adaptivity.

### Zhodnocení dosažených výsledků v 1D

Z provedených experimentů vyplývá, že v adaptivních algoritmech 4.1 a 4.2 je nutné pro výpočet numerického řešení používat taková numerická schémata, která dostatečně přesně zachytí nespojitost, či oblast s velkou hodnotou normy gradientu v numerickém řešení. Dále se ukázalo, že vliv na přesnost výpočtu má i použitý přepočtení řešení. Z hlediska námi použitého ukazatele  $L2E_\rho$ , (4.18) se jako přesnější jeví transformace řešení, jež je založena na perturbační metodě (4.3) s tokem (4.5), než použití váženého průměru (4.10). Z hlediska času potřebného pro přepočtení řešení je ve většině případů dosahováno nižších hodnot při použití váženého průměru (4.10). V uskutečněných experimentech se ukázalo, že použití adaptivity je schopno částečně utlumit vzniklé oscilace poblíž nespojitostí. Domníváme se, že je to díky transformaci řešení.

Dále z provedených experimentů vyplývá, že vhodné nastavení parametrů ovlivňujících adaptivní proces je, vzhledem k větší či menší citlivosti jednotlivých metod na dané parametry, důležité a ovlivňuje získané výsledky. Jako méně



Obrázek 4.7: Porovnání spočtených řešení pro modifikovaný Sodův problém, Lax-Wendroffův numerický tok s limiterem,  $J = 160$ . Nahoře (a),(b) je porovnání numerického řešení získaného pomocí metod *GSMOVE-PM* s  $\alpha = 0.0015$  (a) a *GSVQPM-PM* s  $\varepsilon_1 = 20$ ,  $p = 100$  (b) s přesným řešením v čase  $t = 0.4$ . Dole (c),(d) jsou zachyceny trajektorie jednotlivých uzlů sítě při užití metod *GSMOVE-PM* (c) a *GSVQPM-PM* (d).

citlivá na změnu parametrů se ukázala námi navržená metoda *GS-VQPM*, viz tabulka 4.1.

## 4.4 Numerické experimenty ve 2D

V této části naší práce se zaměříme na otestování výše uvedeného algoritmu<sup>22</sup> 4.1 ve 2D. Jako numerický řešič budeme používat metodu konečných objemů pro řešení Eulerových rovnic (1.2) se schématem ADER, popsáným v kapitole 2.2.1 a kapitole 2.6.5, s přibližným řešičem HLLC, viz 2.4.1. Jako adaptivní modul algoritmu 4.1 použijeme buď variačního přístupu, viz kapitola 3.1.5, nebo naši původní metodu založenou na minimalizaci kvality vrcholu, konkrétně metodu s opravným cyklem, viz algoritmus 3.8 v kapitole 3.3.3. Adaptace sítě je v obou případech prováděna dle první složky lineární rekonstrukce  $\mathbf{w}_h^*$  (3.82) která odpovídá hustotě  $\rho$ . Pro přepočítání řešení ze staré sítě na síť novou je využito jak postupu (4.14), tak i přepočtu pomocí váženého průměru (4.10).

Implementace jednotlivých částí algoritmu 4.1 byla následující. Jako řešič byla použita metoda konečných objemů se schématem ADER, která je implementována ve Fortranu a je součástí práce [23], která je v současné době dále rozšiřována

<sup>22</sup>Protože jsme v provedených 1D numerických experimentech viděli, že použití predikce, algoritmus 4.2, sice zvýší přesnost, ale též čas nyní pro běh programu, nebudeme se jí dále zabývat.

v rámci doktorského studia. Pro adaptaci sítě budeme, v případě metody s opravným cyklem a následného přepočtu řešení, používat naši vlastní implementaci v jazyce C++. V případě aplikace variačních metod budeme používat pro adaptaci sítě naši vlastní implementaci v jazyce FreeFem++ verze 3.40. A pro transformaci řešení pomocí váženého průměru použijeme vlastní implementaci v C++.

## Sodova trubice ve 2D

Pro tento test jsme si vybrali 2D rozšíření úlohy Sodovy trubice popsané v kapitole 4.3. Jako oblast  $\Omega$ , na které řešíme Eulerovy rovnice (1.2), uvažujme oblast  $\Omega = [-3, 3] \times [0, 0.1]$ . Počáteční podmínka je :

$$\mathbf{w}^0(x) = \begin{cases} (1, 0, 0, 5)^T, & x_1 < 0, \\ (1.101463, 0, 0, 2.5)^T, & x_1 > 0, \end{cases} \quad (4.20)$$

a okrajová podmínka je zvolena tak, že levý okraj oblasti je vstup a pravý výstup, dolní a horní okraje oblasti jsou pevné nepropustné stěny, viz okrajové podmínky v [13, str. 203]. Fyzikální čas, do kterého je úloha řešena, volíme  $t = 1$ . Pro podmínku stability bylo voleno  $CFL$  číslo  $CFL = 0.3$ , jak při použití adaptivity, tak bez ní.

Z toho důvodu, že jednotlivé moduly algoritmu 4.1 jsou nyní implementovány v různých jazycích a výměnu dat mezi nimi realizujeme pomocí souborových operací, tak pro metodu adaptace sítě s opravným cyklem provádíme její adaptaci a následný přepočet numerického řešení, ze staré sítě na novou, nikoliv v každém časovém okamžiku  $t_k$ , nýbrž pro fyzikální čas  $t = j\Delta t$ ,  $j = 1, 2, \dots$ , kde  $\Delta t = 0.01$ . V případě použití metod adaptace sítě založených na variačním principu budeme adaptovat pro fyzikální čas  $t = j\Delta t$ ,  $j = 1, 2, \dots$ , kde  $\Delta t = 0.1$ . Důvodem je to, že při adaptaci sítě tímto způsobem dojde ke změně poloh všech uzlů sítě najednou, viz algoritmus 3.1, tato posunutí navíc nemusí být malá, proto nelze použít perturbační metodu pro přepočet řešení a z toho důvodu je nutné použít transformaci řešení založenou na váženém průměru, viz vztah (4.10), která velmi zvyšuje časovou náročnost výpočtu.

Opět jako v předchozích experimentech budeme sledovat chybu vůči přesnému řešení ve složce numerického řešení  $\mathbf{w}$ , které odpovídá hustotě  $\rho_h$  v časovém kroku  $t_k$ :

$$L1E_\rho = \int_{\Omega} |\rho(\mathbf{x}, t_k) - \rho_h(\mathbf{x}, t_k)| dx \approx \sum_{D_i \in \mathcal{T}_h} |D_i| |\rho(\mathbf{x}_i, t_k) - \rho_h(\mathbf{x}_i, t_k)|, \quad (4.21)$$

resp.

$$L2E_\rho = \sqrt{\int_{\Omega} (\rho(\mathbf{x}, t_k) - \rho_h(\mathbf{x}, t_k))^2 dx} \approx \sqrt{\sum_{D_i \in \mathcal{T}_h} |D_i| (\rho(\mathbf{x}_i, t_k) - \rho_h(\mathbf{x}_i, t_k))^2}, \quad (4.22)$$

kde  $\mathbf{x}_i$  je těžiště objemu  $D_i$ , jako přesné řešení  $\rho$  v těžišti  $\mathbf{x}_i$  objemu  $D_i$  je použito řešení uvedené v [13, str. 137].

Pro metodu založenou na variačním principu bylo použita monitorovací funkce Winslowova typu (3.43) a nastavení, které je uvedeno v tabulce 4.11. Výpočetní oblast  $\Omega_c$  jsme volili  $\Omega_c = \Omega$  a síť na této oblasti odpovídala síti na oblasti  $\Omega$  bez použití adaptivity.

Parametr	Hodnota	Popis
$\Delta\tau$	0.001	velikost časového kroku při užití metody umělého času (3.64)
$\mathcal{T}$	0.01	parametr rychlosti adaptace sítě (3.59)
$\epsilon$	$10^{-6}$	minimální hodnota ukazatele dosažení stac. stavu <i>REZ</i> (3.71)
<i>TSTEP</i>	10	počet iterací metody pro dosažení stacionárního stavu
<i>MAXITER</i>	1	počet provedených adaptací sítě
<i>MAXITER<sub>b</sub></i>	10	počet iterací schématu (3.58) pro určení poloh hraničních uzlů

Tabulka 4.11: Použité parametry pro metodu založenou na variačním principu, viz algoritmus 3.1 v kapitole 3.1.5.

Pro metodu s opravným cyklem bylo použito nastavení, které je uvedeno v tabulce 4.12.

Parametr	Hodnota	Popis
$\alpha$	0.02	minimální úhel, viz (3.111)
$\gamma_r$	0.9	multiplikátor, viz (3.126)
<i>MAXITG</i>	5	počet iterací metody zlatého řezu
<i>MAXITREP</i>	5	počet opravných iterací
<i>MAXITGRAD</i>	1	počet iterací gradientní metody nad jedním uzlem
<i>MAXITER</i>	1	počet globálních iterací

Tabulka 4.12: Použité parametry pro metodu s opravným cyklem, viz algoritmy 3.5, 3.6, 3.7 a 3.8.

Pro porovnání jsou v tabulce 4.13 zaneseny některé sledované veličiny pro různý počet objemů  $J$  bez použití adaptivity, kde *ITER* představuje počet kroků metody konečných objemů.

$J$	$L1E_\rho$	$L2E_\rho$	<i>ITER</i>	$T$ [s]
320	0.007455	0.030829	377	0.88
548	0.005919	0.028377	591	2.38
960	0.004945	0.025725	1012	7.46
3840	0.003189	0.021283	1604	49.48
6090	0.002728	0.019303	2489	116.97

Tabulka 4.13: Sledované veličiny pro Sodův problém v  $t = 1$ , pro numerické schéma ADER, bez použití adaptivity.

V následující tabulce 4.14 jsou zaneseny sledované parametry pro Sodův problém ve 2D, za použití variačního přístupu s Winslowovou monitorovací funkcí, pro různé volby parametru<sup>23</sup>  $\alpha$ , viz (3.43), a různé počty objemů  $J$ , kde  $h_{min}$ , resp.  $h_{max}$  představují délky nejkratší, resp. nejdelší hrany sítě  $\mathcal{T}_h$  a  $\epsilon$  je jejich poměr na konci výpočtu, čili ukazatel míry adaptace.

V tabulce 4.15 jsou zaneseny sledované parametry pro Sodův problém ve 2D, za použití adaptivní metody s opravným cyklem, pro různé volby parametrů  $\epsilon_1$  a  $p$ , viz ((3.108) a (3.107)).

<sup>23</sup>V případě  $J = 320$  objemů byla adaptace sítě pro  $\alpha = 0.25$  a  $\alpha = 0.5$  minimální, z toho důvodu byly použity jiné hodnoty parametru  $\alpha$  než v dalších experimentech.

$J$	$\alpha$	$h_{min}$	$h_{max}$	$\varepsilon$	$L1E_\rho$	$L2E_\rho$	$T$ [s]
320	0.8	0.019635	0.100789	5.13	0.007230	0.030039	24.41
320	0.9	0.017557	0.107007	6.09	0.007196	0.029557	24.62
320	1.0	0.015351	0.116751	7.61	0.007135	0.028962	25.43
548	0.25	0.017283	0.092798	5.37	0.005689	0.027030	44.91
548	0.5	0.011567	0.091715	7.93	0.005650	0.025726	47.46
548	0.75	0.010756	0.089330	8.30	0.006039	0.027086	48.71
960	0.25	0.014743	0.067475	4.58	0.004680	0.023220	78.87
960	0.5	0.009466	0.068043	7.19	0.004560	0.022290	90.07
960	0.75	0.009321	0.079031	8.48	0.004696	0.022823	93.17
3840	0.25	0.006706	0.033956	5.06	0.003516	0.020127	698.07
3840	0.5	0.006319	0.034789	5.51	0.003692	0.021238	670.97
3840	0.75	0.005465	0.034821	6.37	0.003579	0.021308	631.29

Tabulka 4.14: Sledované veličiny pro 2D Sodův problém v  $t = 1$ , pro numerické schéma ADER v kombinaci s variačním přístupem.

$J$	$\varepsilon_1$	$p$	$h_{min}$	$h_{max}$	$\varepsilon$	$L1E_\rho$	$L2E_\rho$	$T$ [s]
320	100	10	0.018523	0.223870	12.09	0.006671	0.029825	10.88
320	200	10	0.014007	0.253834	18.12	0.006585	0.029105	11.14
320	500	10	0.010790	0.299429	27.75	0.006680	0.028934	11.65
548	100	10	0.011612	0.152088	13.10	0.005538	0.027538	24.89
548	200	10	0.010458	0.183396	17.54	0.005445	0.027263	32.48
548	500	10	0.007338	0.212749	28.99	0.005376	0.026442	42.78
960	100	10	0.009397	0.149933	15.96	0.004713	0.024444	40.69
960	200	10	0.008582	0.183681	21.40	0.004617	0.024182	42.92
960	500	10	0.007972	0.217308	27.26	0.004522	0.023955	48.35
3840	100	10	0.006779	0.059302	8.75	0.002979	0.020085	274.17
3840	200	10	0.004615	0.063899	13.84	0.003026	0.019832	393.61
3840	500	10	0.004385	0.089610	20.44	0.002935	0.019152	428.12

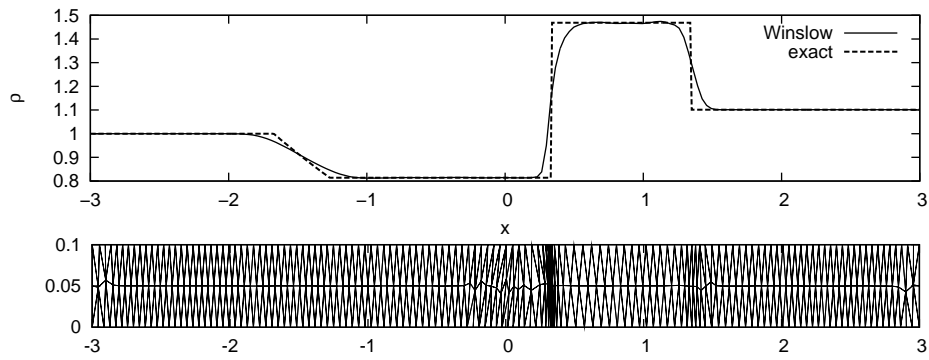
Tabulka 4.15: Sledované veličiny pro 2D Sodův problém v  $t = 1$ , pro numerické schéma ADER v kombinaci s metodou s opravným cyklem.

Ze srovnání získaných výsledků je vidět, že v případě použití metody založené na variačním principu došlo v několika případech ke zhoršení v ukazateli  $L1E_\rho$  vůči řešení bez použití adaptivity, zatímco z hlediska ukazatele  $L2E_\rho$  došlo ke zlepšení. Dále je vidět, že výsledky získané za použití metody založené na variačním principu jsou horší, než za použití adaptivity pomocí metody s opravným cyklem, kde vedlo použití adaptivity ke zvýšení přesnosti výpočtu v obou sledovaných ukazatelích  $L1E_\rho$  a  $L2E_\rho$ . Domníváme se, že je to způsobeno tím, že v obou testech používáme jiný přepočítání řešení ze staré sítě na novou adaptovanou síť a tento způsob přepočtu, tj. použití váženého průměru, dosahoval i v provedených 1D experimentech horších výsledků z hlediska použitých ukazatelů.

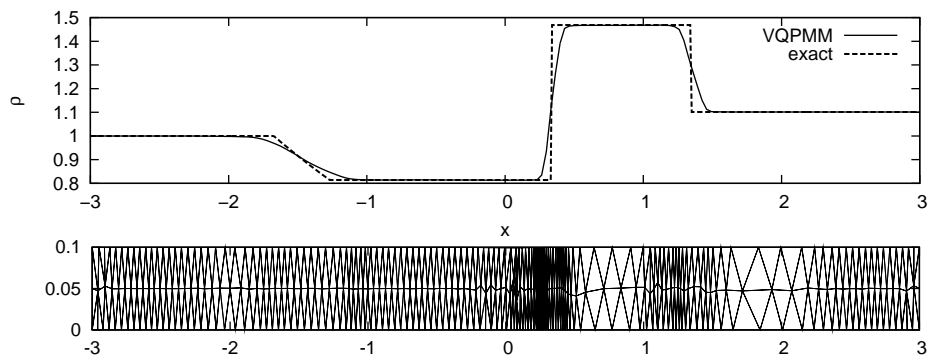
Na obrázcích 4.8, 4.9, 4.10 a 4.11 je zaneseno porovnání spočtených hodnot s přesným řešením na spodní hraně oblasti  $\Omega$  a dále jsou zde zakresleny výsledné sítě. Zde je vidět, že zjemnění sítě odpovídá místům nespojitosti a opět, jako v 1D, je struktura zjemněné sítě pro každou adaptivní strategii jiná. Metoda založená na variačním principu (Winslowova) zachycuje nespojitost ostřeji, ale

v místech s konstantní hodnotou sledované veličiny je adaptace sítě minimální, na rozdíl od metody založené na minimalizaci parametru kvality vrcholu, která síť zjemní v místech nespojitosti a v místech s konstantní hodnotou sledované veličiny dochází k zhrubnutí sítě, opět obdobně jako v 1D.

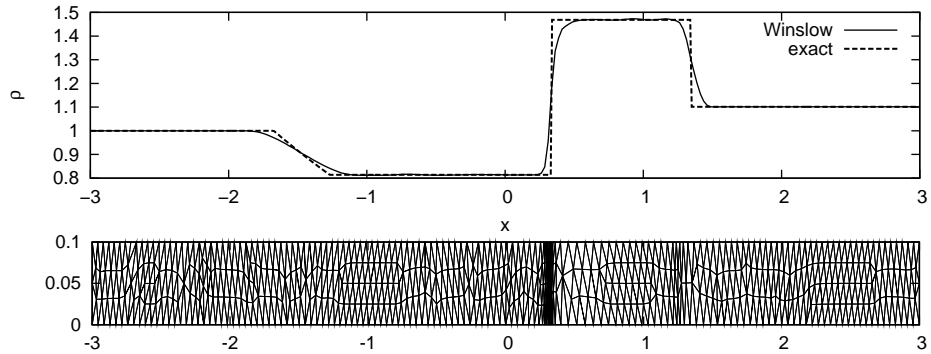
Na obrázcích 4.12 je zachycen detail porovnání získaného numerického řešení s použitím adaptivity a bez ní s přesným řešením. Zde je vidět, že ve všech případech dochází k ostřejšímu zachycení nespojitosti řešení, tj. při použití adaptivity je norma gradientu hustoty  $\rho$  v okolí nespojitosti větší, než bez ní. Pro metodu založenou na variačním principu (Winslowova) dochází za nespojitostí k pomalejšímu přimykání numerického řešení k přesné hodnotě, než v případě bez použití adaptivity. Toto chování je nejspíše způsobeno použitým přepočtem řešení, tj. použitím váženého průměru. A je též příčinou toho, že tato metoda dosahuje horších výsledků v sledovaných ukazatelích  $L1E_\rho$  a  $L2E_\rho$  než druhá metoda, kde se používá pro přepočet řešení perturbační metoda, čili vztahu (4.14).



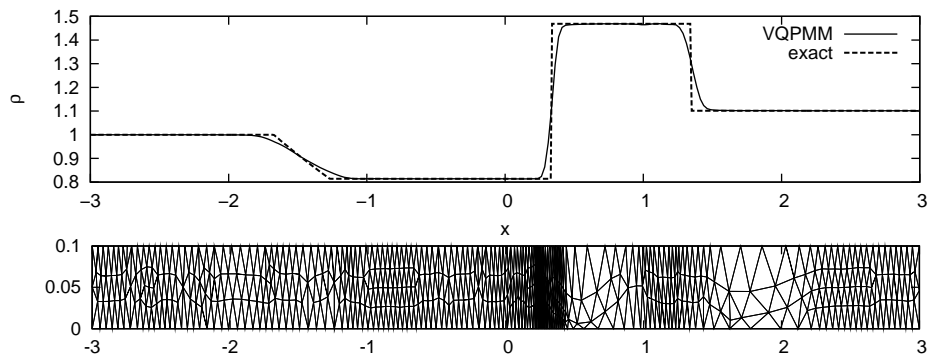
Obrázek 4.8: Porovnání spočteného řešení pro Sodovu trubici s přesným řešením (nahore) a výsledná síť (dole),  $J = 960$  objemů,  $\alpha = 0.75$  za použití adaptivity, Winslowova metoda.



Obrázek 4.9: Porovnání spočteného řešení pro Sodovu trubici s přesným řešením (nahore) a výsledná síť (dole),  $J = 548$  objemů,  $\varepsilon_1 = 500$ ,  $p = 10$  za použití adaptivity, metoda s opravným cyklem.

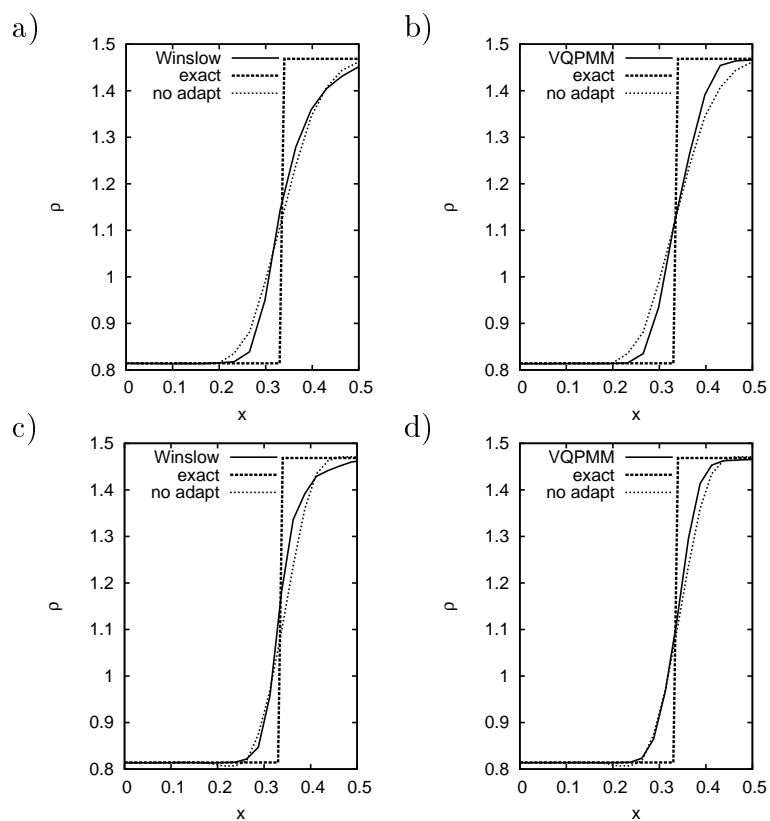


Obrázek 4.10: Porovnání spočteného řešení pro Sodovu trubici s přesným řešením (nahore) a výsledná síť (dole),  $J = 960$  objemů,  $\alpha = 0.75$  za použití adaptivity, Winslowova metoda.



Obrázek 4.11: Porovnání spočteného řešení pro Sodovu trubici s přesným řešením (nahore) a výsledná síť (dole),  $J = 960$  objemů,  $\varepsilon_1 = 500$ ,  $p = 10$  za použití adaptivity, metoda s opravným cyklem.





Obrázek 4.12: Detail zachycení nespojitosti pro testované metody. Nahoře (a),(b) pro  $J = 548$  objemů: Winslowova metoda,  $\alpha = 0.75$  (a), metoda s opravným cyklem,  $\varepsilon_1 = 500$ ,  $p = 10$  (b). Dole (c),(d) pro  $J = 960$  objemů: Winslowova metoda,  $\alpha = 0.75$  (c), metoda s opravným cyklem,  $\varepsilon_1 = 500$ ,  $p = 10$  (d).

## Zhodnocení provedených experimentů ve 2D

Z provedených experimentů vyplývá, že při použití přepočtu řešení pomocí váženého průměru (4.10) se dosahuje, z hlediska sledovaných ukazatelů  $L1E_\rho$  (4.21) a  $L2E_\rho$  (4.22), horších výsledků než v případě použití perturbační metody, a navíc je tento způsob časově mnohem náročnější.

U metod založených na variačním principu používáme, v případě trojúhelníkové sítě konečných objemů, pro řešení systému rovnic (3.2) určujících polohy uzlů metodu konečných prvků, a tak získáme polohy všech uzlů sítě naráz. Z toho důvodu není možné použít pro přepočet perturbační metodu, kde se předpokládá vždy pohyb jednotlivých uzlů sítě. Poznamenejme, že zde se lišíme od práce [40], kde byl algoritmus typu 4.1 aplikován na čtyřúhelníkové síti a pro řešení systému rovnic (3.2) byla použita metoda sítí. Iterace metody sítí byly prováděny nad jednotlivými uzly spolu s přepočtem řešení založeným na perturbační metodě.

Při pokusu implementovat posun jednotlivých uzlů tak, abychom byli schopni použít perturbační metodu, se ukázalo jako nutné použít velmi malé hodnoty  $\alpha$  v monitorovací funkci (3.43). To ale způsobilo, že adaptace sítě byla minimální, proto jsme se tím dále nezabývali. Důvodem k volbě malé hodnoty  $\alpha$  je podmínka malého posunu uzlů pro aplikaci přepočtu řešení pomocí perturbační metody.

Při porovnání obou postupů se, v případě trojúhelníkové sítě konečných objemů, jeví jako výhodnější použití metody založené na minimalizaci parametru kvality uzlu, která je schopna jak zhuštění uzlů v okolí nespojitosti či oblasti s velkou normou gradientu řešení, tak i opačného procesu v oblastech, kde je řešení téměř konstantní. Navíc lze u této metody použít pro přepočet numerického řešení perturbační metody, která dosahuje lepších výsledků. Poznamenejme, že námi použitý přepočet pomocí perturbační metody (4.14) využívá pouze hodnotu po částech konstantního numerického řešení na daném objemu či jeho sousedech, v čemž se lišíme od 1D úlohy, kde byla použita lineární rekonstrukce (4.5) hodnot na hranicích objemu.

# Závěr

V práci jsme se zabývali problémem adaptace výpočetní sítě v kontextu metody konečných objemů pro řešení Eulerových rovnic, popisujících proudění nevazké stlačitelné tekutiny.

Popsali jsme matematický model proudění nevazké stlačitelné tekutiny a metodu konečných objemů, užívanou pro jeho řešení, se zaměřením na numerické toky, jež byly užity v našich numerických experimentech. Dále jsme popsali některé vybrané adaptivní strategie, ze kterých naše metody vycházejí, nebo s nimi naše metody srovnáváme a tyto jsme zároveň otestovali. Popsali jsme metody přepočtu numerického řešení Eulerových rovnic mezi původní a nově adaptovanou sítí konečných objemů, které splňují geometrický zákon zachování a jsou vhodné při použití s adaptací sítě konečných objemů v 1D a ve 2D. Formulovali a otestovali jsme adaptační algoritmus pro řešení úloh s pohybující se nespojitostí, či s oblastí s velkou hodnotou normy gradientu některé ze složek numerického řešení.

V práci jsme dosáhli následujících **původních** výsledků. V 1D jsme navrhli vlastní metodu adaptace sítě vycházející z anisotropní adaptivity, která je rychlostí a přesností výpočtu, měřenou ve vhodném ukazateli, srovnatelná s již používanou strategií tzv. Moving Mesh algoritmu [40]. Je méně citlivá na volbu parametrů ovlivňujících adaptační proces, což jsme ověřili na některých numerických experimentech. Dále jsme v 1D prověřili chování adaptačního procesu v závislosti na použitém numerickém řešiči Eulerových rovnic a postupu přepočtu řešení mezi původní a adaptovanou sítí.

Pro úlohy ve 2D jsme navrhli vlastní adaptační strategie, vycházející z anisotropní adaptivity a tyto jsme prověřili na schopnost adaptovat síť za různých podmínek kladených na výpočetní oblast. Dále jsme ověřili možnosti použití těchto adaptačních strategií v kombinaci s numerickým schematem metody konečných objemů ADER při řešení Eulerových rovnic ve 2D na úloze s pohybující se nespojitostí.

Pro trojúhelníkovou síť ve 2D jsme formulovali metodu přepočtu numerického řešení Eulerových rovnic mezi původní a nově adaptovanou sítí konečných objemů, jež splňuje geometrický zákon zachování.

Do budoucna je nutné se zaměřit na navržení postupu dynamického škálování parametrů ovlivňujících proces adaptace tak, aby byl co nejvíce automatizovaný během výpočtu. Pro úlohy ve 2D bude klíčové věnovat pozornost strategii přepočtu řešení při výrazných změnách sítě či její topologie a navrhnout strategii přepočtu řešení “vyššího řádu”, obdobnou strategií užitou v 1D.

Protože se naše původní metoda adaptivního zjemnění sítě ukázala srovnatelná a v řadě aspektů lepší než metoda založená na variačním principu, předpokládáme její další vývoj v kontextu metod konečných objemů vyšších řádů s aplikacemi nejen pro řešení Eulerových rovnic, ale i pro řešení dalších hyperbolických rovnic, se kterými se setkáváme při řešení rovnic mělké vody [43] nebo sedimentace polydisperzních směsí [2].

# Literatura

- [1] BAECK, T.; FOGEL, D. B.; MICHALEWICZ, Z.: *Evolutionary Computation 2*. Institute of Physics Publishing, 2000, ISBN 9780750306652.
- [2] BERRES, S.; BURGER, R.; KARLSEN, K. H.; aj.: Strongly Degenerate Parabolic-Hyperbolic Systems Modeling Polydisperse Sedimentation with Compression. *SIAM Journal on Numerical Analysis*, 2003.
- [3] BOYD, S.; VANDENBERGHE, L.: *Convex Optimization*. Cambridge University Press, 2009, ISBN 978-0-521-83378-3.
- [4] BRACKBILL, J. U.: An Adaptive Grid with Directional Control. *Journal of Computational Physics* 108, 2001.
- [5] CAO, W.; HUANG, W.; RUSSELL, R. D.: An r-Adaptive Finite Element Method Based upon Moving Mesh PDEs. *Journal of Computational Physics* 149, 1999: s. 221–244.
- [6] CAO, W.; HUANG, W.; RUSSELL, R. D.: A Study Of Monitor Function For Two-Dimensional Adaptive Mesh Generation. *SIAM Journal on Numerical Analysis*, ročník 20, č. 6, 1999: str. 1978–1994.
- [7] CASTILLO, J. E.: *Mathematical Aspects of Numerical Grid Generation*. Society for Industrial and Applied Mathematics, 1991, ISBN 9780898712674.
- [8] CENICEROS, H. D.; HOU, T. Y.: An Efficient Dynamically Adaptive Mesh for Potentially Singular Solutions [online]. *Journal of Computational Physics*, 2001: s. 609–639, [cit. 2011-5-3].  
Dostupné z URL: <<http://www.acm.caltech.edu/~hou/papers/adaptive.pdf>>
- [9] DOLEJŠÍ, V.: *Manuál programu ANGENER 3.1 [online]*. [cit. 2011-5-3].  
Dostupné z URL: <<http://www.karlin.mff.cuni.cz/~dolejsi/angen/angen3.1.htm>>
- [10] DOLEJŠÍ, V.: Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Computing and Visualisation in Science*, 1998: s. 165–178.
- [11] FEISTAUER, M.: *Mathematical methods in fluid dynamics*. Chapman and Hall/CRC, 1993, ISBN 0582209889.
- [12] FEISTAUER, M.; FELCMAN, J.; Lukáčová-Medvid'ová, M.: Combined finite element-finite volume solution of compressible flow. *J. Comput. Appl. Math*, 1995: s. 179–199.
- [13] FEISTAUER, M.; FELCMAN, J.; STRAŠKRABA, I.: *Mathematical and Computational Methods for Compressible Flow*. Oxford University Press, Oxford, 2003, ISBN 9780198505884.
- [14] FELCMAN, J.: Počítačové modelování [online]. [cit. 2011-5-3].  
Dostupné z URL: <[http://artax.karlin.mff.cuni.cz/~nozkj1am/mff/MOD104/\\_MatMod.pdf](http://artax.karlin.mff.cuni.cz/~nozkj1am/mff/MOD104/_MatMod.pdf)>

- [15] FELCMAN, J.; KUBERA, P.: Computational Aspects of the Mesh Adaptation. In *Proceedings of the colloquium Dynamika Tekutin 2004.*, Institute of Thermomechanics AS CR, 2004, ISBN 80-85918-89-7.
- [16] FELCMAN, J.; KUBERA, P.: Computational Aspects of the Mesh Adaptation for the Time Marching Procedure. In *Proceedings of the ENUMATH 2005 conference*, Springer Berlin Heidelberg New York, 2005, ISBN 978-3-540-34287-8.
- [17] FELCMAN, J.; KUBERA, P.: An Adaptive Mesh Method for Hyperbolic Conservation Laws. In *Proceedings of the Fifth International Conference on Engineering Computational Technology, ECT 2006*, Civil-Comp Press, Stirlingshire, 2006, ISBN 1-905088-10-8, cDROM.
- [18] FELCMAN, J.; KUBERA, P.: A higher-order adaptive method for the Euler equations. In *International Conference on Numerical Analysis and Applied Mathematics 2006, ICNAAM 2006*, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2006, s. 650–653.
- [19] FELCMAN, J.; KUBERA, P.: A mesh adaptation for non-stationary problems. In *Sixth International Congress on Industrial Applied Mathematics (ICIAM07) and GAMM Annual Meeting, 7*, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2007, str. 2100007–2100008.
- [20] FELCMAN, J.; KUBERA, P.: Vertex Quality Parameter Moving Mesh Method. In *Numerical Analysis and Applied Mathematics, ICNAAM 2007, AIP Conference Proceedings*, 936, American Institute of Physics, 2007, s. 204–207.
- [21] FELCMAN, J.; KUBERA, P.: An Adaptive Finite Volume Method for Non-stationary Problems. In *Finite Volumes for Complex Applications V, Problems & Perspectives*, ISTE Ltd and John Wiley and Sons, Inc, 2008, ISBN 978-1-84821-035-6, s. 407–415.
- [22] FELCMAN, J.; KUBERA, P.: A Comparison of Adaptation Techniques for the Solution of Nonstationary Flow. In *International Conference of Numerical Analysis and Applied Mathematics, ICNAAM 08, AIP Conference Proceedings*, American Institute of Physics, 2008, s. 835–838.
- [23] FINDEJS, K.: *Kombinovaná metoda konečných objemů a konečných prvků pro řešení 3D proudění*. Diplomová práce, Matematicko-fyzikální fakulta, Univerzita Karlova v Praze, 2005.
- [24] HECHT, F.; PIRONNEAU, O.; HYARIC, A. L.; aj.: *Manual FreeFem++ [online]*. [cit. 2011-5-3].  
Dostupné z URL: <<http://www.freefem.org>>
- [25] HUANG, W.; REN, Y.; RUSSELL, R. D.: Moving Mesh Methods Based on Moving Mesh Partial Differential Equations. *Journal of Computational Physics*, 1995: str. 28.
- [26] HUANG, W.; RUSSEL, R. D.: *Adaptive Moving Mesh Methods*. Springer, 2010, ISBN 978-1-4419-7915-5.
- [27] HUANG, W.; RUSSELL, R. D.: Moving Mesh Strategy Based On Gradient Flow Equation For Two-Dimensional Problems. *SIAM Journal on Numerical Analysis*, ročník 20, 1999: s. 998–1015.

- [28] HUDSON, J.: A Review on the Numerical Solution of the 1D Euler Equations. 2006, manchester Institute for Mathematical Sciences, School of Mathematics, The University of Manchester.
- [29] KADRŇKA, L.: *Schéματα typu Moving Mesh pro řešení nestacionárních úloh*. 2007, Bakalářská práce, Matematicko-fyzikální fakulta, Univerzita Karlova v Praze.
- [30] KUBERA, P.: On a mesh adaptation for nonstationary problems. In *WDS'04 Proceedings of Contributed Papers: Part III - Physics*, editace J. Šafránková, Praha, Matfyzpress, 2004, s. 613–619.
- [31] KURGANOV, A.; TADMOR, E.: New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations. *Journal of Computational Physics*, 2000: s. 241–282.
- [32] LUKŠAN, L.: *Numerické optimalizační metody pro úlohy bez omezujících podmínek [online]*. Akademie věd České republiky, 1995, učební text V-640. [cit. 2011-5-3]. Dostupné z URL: <ftp://ftp.cs.cas.cz/pub/reports/v640-95.ps>
- [33] MACKENZIE, J. A.; STOCKIE, J. M.; RUSSELL, R. D.: A Moving Mesh Method for One-Dimensional Hyperbolic Conservation Laws. *SIAM Journal on Numerical Analysis*, ročník 22, 2001: s. 1791–1813.
- [34] MÜLLER, B.; REINHARDT, J.; STRICKLAND, M. T.: *Neural Networks: an introduction*. Springer, 1995, ISBN 3540602070.
- [35] PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; AJ.: *Numerical Recipes in C*. Cambridge University Press, druhé vydání vydání, 1994, ISBN 0521431085.
- [36] QUARTERONI, A.; SACCO, R.; SALERI, F.: *Numerical Mathematics*. Springer, 2007, ISBN 978-3-540-34658-6.
- [37] QUEIROZ, R. A. B.; FERREIRA, V. G.; CUENCA, R. G.: New High Resolution TVD Scheme for Unsteady Flows with Shock Waves [online]. In *Selected papers from the 7th Brazilian Conference on Dynamics, Control and Applications - DINCON 2008*, 2008, [cit. 2011-5-3]. Dostupné z URL: [http://www.sbmac.org.br/tema/seletas/docs/v9\\_2/14\\_Que\\_Fer\\_Cue.pdf](http://www.sbmac.org.br/tema/seletas/docs/v9_2/14_Que_Fer_Cue.pdf)
- [38] TADMOR, E.: Numerical Viscosity and the Entropy Condition for Conservative Difference Schemes. *Mathematics of Computation*, ročník 43, č. 168, 1984: s. 369–381.
- [39] TANG, H.: A Moving Mesh Method for the Euler Flow Calculations Using a Directional Monitor Function. *Communications in Computational Physics*, ročník 1, č. 4, 2006: s. 656–676.
- [40] TANG, H.; TANG, T.: Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, ročník 41, č. 2, 2003: str. 487–515.
- [41] TIMMES, F. X.: Zdrojový kód programu [online], exact\_riemann.f, [cit. 2011-5-3]. Dostupné z URL: [http://cococubed.asu.edu/code\\_pages/exact\\_riemann.shtml](http://cococubed.asu.edu/code_pages/exact_riemann.shtml)

- [42] TITAREV, V. A.: *Derivative Riemann problem and ADER schemes*. Dizertační práce, University of Trento, Itálie, 2005.
- [43] TORO, E. F.: *Shock-capturing Methods for Shallow Flows*. Wiley-Blackwell, 2001, ISBN 9780471987666.
- [44] TORO, E. F.; DUMBSER, M.; TITAREV, V. A.; aj.: The Derivative Riemann Problem: The Basis for High Order Ader Schemes. In *European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006*, 2006, str. 20.
- [45] TORO, E. F.; TITAREV, V. A.: Ader schemes for scalar hyperbolic conservation laws in three space dimensions [online]. Preprint NI03063, 2004, [cit. 2011-5-3]. Dostupné z URL: <<http://www.newton.ac.uk/preprints/NI03063.pdf>>
- [46] TORO, E. F.; TITAREV, V. A.: Ader schemes for Three-Dimensional Non-Linear Hyperbolic System [online]. Preprint NI04007, 2004, [cit. 2011-5-3]. Dostupné z URL: <<http://www.newton.ac.uk/preprints/NI04007.pdf>>
- [47] ZHANG, Z.: Moving Mesh Method with Conservative Interpolation Based on L2-Projection. *Communications in Computational Physics*, ročník 1, č. 5, 2006: s. 930–944.
- [48] ŠOLÍN, P.; FELCMAN, J.: On the Construction of the Osher-Solomon Scheme for the 3D Euler Equations. 1999, East-West J. Numer. Math.

# Seznam tabulek

3.1	Sledované veličiny, úloha A v 1D, použití umělého času. . . . .	55
3.2	Sledované veličiny, úloha A v 1D, použití G-S iterací. . . . .	56
3.3	Sledované veličiny, úloha B v 1D, použití umělého času. . . . .	58
3.4	Sledované veličiny, úloha B v 1D, použití G-S iterací. . . . .	58
3.5	Sledované veličiny, úloha A ve 2D, Winslowova metoda. . . . .	62
3.6	Sledované veličiny, úloha A ve 2D, monitorovací funkce ALLMF. . . . .	62
3.7	Sledované veličiny, úloha B ve 2D, použita Winslowova metoda. . . . .	64
3.8	Sledované veličiny, úloha B ve 2D, použita Winslowova metoda. . . . .	65
3.9	Sledované veličiny, úloha B ve 2D, použita Winslowova metoda. . . . .	65
3.10	Sledované veličiny, úloha B ve 2D, monitorovací funkce ALLMF. . . . .	67
3.11	Sledované veličiny, úloha B ve 2D, monitorovací funkce ALLMF. . . . .	67
3.12	Sledované veličiny, úloha B ve 2D, monitorovací funkce ALLMF. . . . .	68
3.13	Sledované veličiny, úloha B ve 2D, funkce ALLMF s normováním. . . . .	69
3.14	Sledované veličiny, úloha B ve 2D, funkce ALLMF s normováním. . . . .	69
3.15	Sledované veličiny, úloha B ve 2D, funkce ALLMF s normováním. . . . .	69
3.16	Sledované veličiny, úloha A ve 2D, použita Winslowova metoda. . . . .	72
3.17	Sledované veličiny, úloha A ve 2D, monitorovací funkce ALLMF. . . . .	72
3.18	Sledované veličiny, úloha B ve 2D, použita Winslowova metoda. . . . .	73
3.19	Sledované veličiny, úloha B ve 2D, funkce ALLMF s normováním. . . . .	73
3.20	Sledované veličiny, úloha A v 1D, Jacobiova metoda. . . . .	95
3.21	Sledované veličiny, úloha A v 1D, Gaussova-Seidelova metoda. . . . .	95
3.22	Sledované veličiny, úloha B v 1D, Jacobiova metoda. . . . .	99
3.23	Sledované veličiny, úloha B v 1D, Gaussova-Seidelova metoda. . . . .	99
3.24	Použité parametry, metoda s bariérovou funkcí. . . . .	102
3.25	Použité parametry, metoda s kontrolou úhlů. . . . .	103
3.26	Použité parametry, metoda s opravným cyklem. . . . .	103
3.27	Sledované veličiny pro metodu s bariérovou funkcí, A ve 2D. . . . .	103
3.28	Sledované veličiny pro metodu s kontrolou úhlů, A ve 2D. . . . .	104
3.29	Sledované veličiny pro metodu s opravným cyklem, A ve 2D. . . . .	104
3.30	Sledované veličiny pro metodu s bariérovou funkcí, úloha B ve 2D. . . . .	107
3.31	Sledované veličiny pro metodu s kontrolou úhlů, úloha B ve 2D. . . . .	107
3.32	Sledované veličiny pro metodu s opravným cyklem, úloha B ve 2D. . . . .	107
3.33	Sledované veličiny pro úlohu s nelineární hranicí. . . . .	111
4.1	Popis testovaných metod. . . . .	121
4.2	Sledované veličiny pro Burgersovu rovnici, metody bez predikce. . . . .	122
4.3	Sledované veličiny pro Burgersovu rovnici, metody s predikcí. . . . .	122
4.4	Sledované veličiny pro Sodovu trubici. . . . .	128
4.5	Sledované veličiny pro Sodovu trubici. . . . .	129
4.6	Sledované veličiny pro Sodovu trubici. . . . .	130
4.7	Sledované veličiny pro Sodovu trubici. . . . .	131
4.8	Sledované veličiny pro modifikovaný Sodův problém. . . . .	133
4.9	Volba konstanty $\alpha$ při dynamickém škálování. . . . .	133
4.10	Sledované veličiny pro modifikovaný Sodův problém. . . . .	134
4.11	Použité parametry, variační metoda. . . . .	137
4.12	Použité parametry, metoda s opravným cyklem. . . . .	137
4.13	Sledované veličiny pro Sodův problém ve 2D. . . . .	137
4.14	Sledované veličiny pro Sodův problém ve 2D. . . . .	138
4.15	Sledované veličiny pro Sodův problém ve 2D. . . . .	138



# Seznam obrázků

2.1	Trojúhelníková síť . . . . .	14
2.2	Čtyřúhelníková síť . . . . .	14
2.3	Duální síť . . . . .	14
2.4	Barycentrická síť . . . . .	14
2.5	Struktura řešení Riemannova problému. . . . .	20
2.6	Aproximace struktury řešení Riemannova problému, HLLC. . . . .	21
2.7	Schéma konstrukce hodnot $u_i^{k,\pm}$ . . . . .	27
3.1	Posunutí izokřivek řešení eliptické úlohy. . . . .	39
3.2	Komprese izokřivek . . . . .	39
3.3	Expanze izokřivek . . . . .	39
3.4	Ilustrace vlivu komprese a expanze izokřivek. . . . .	41
3.5	Průběh $\lambda_1$ a $\frac{\partial \lambda_1}{\partial \mathbf{v}_1}$ na $[0, 1] \times [0, 1]$ . . . . .	42
3.6	Vliv poměru $\lambda_1/\lambda_2 = 2, 1, 0.5, 0.25$ na výslednou síť. . . . .	43
3.7	Průběh $\lambda_1$ a $\frac{\partial \lambda_1}{\partial \mathbf{v}_1}$ na $[0, 1] \times [0, 1]$ . . . . .	44
3.8	Ukázka vlivu expanze izokřivek na výslednou síť. . . . .	44
3.9	Průběh funkce $u$ na oblasti $\Omega$ . . . . .	52
3.10	Původní síť v $\Omega$ a její obraz v $\Omega_c$ . . . . .	53
3.11	Výsledná síť na oblasti $\Omega$ . . . . .	53
3.12	Průběhy funkcí $u_A(x)$ a $u_B(x)$ na $[-1, 1]$ . . . . .	54
3.13	Úloha A v 1D, rekonstrukce $u_A^*(x)$ . . . . .	56
3.14	Úloha A v 1D, průběh L2E. . . . .	57
3.15	Úloha A v 1D, průběh REZ. . . . .	57
3.16	Úloha A v 1D, trajektorie jednotlivých uzlů sítě. . . . .	57
3.17	Úloha B v 1D, rekonstrukce $u_B^*(x)$ . . . . .	59
3.18	Úloha B v 1D, průběh L2E. . . . .	59
3.19	Úloha B v 1D, průběh REZ. . . . .	59
3.20	Úloha B v 1D, trajektorie jednotlivých uzlů sítě. . . . .	60
3.21	Průběh funkce $u_A$ ve 2D. . . . .	61
3.22	Průběh funkce $u_B$ ve 2D. . . . .	61
3.23	Úloha typu A ve 2D, průběh ukazatele REZ. . . . .	62
3.24	Úloha typu A ve 2D, výsledné sítě. . . . .	63
3.25	Úloha typu A ve 2D, výsledné sítě. . . . .	63
3.26	Úloha typu A ve 2D, výsledné sítě. . . . .	64
3.27	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	64
3.28	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	66
3.29	Úloha typu B ve 2D, výsledné sítě. . . . .	66
3.30	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	67
3.31	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	68
3.32	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	69
3.33	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	70
3.34	Úloha typu B ve 2D, výsledné sítě. . . . .	70
3.35	Úloha typu A ve 2D, trajektorie uzlů. . . . .	71
3.36	Úloha typu A ve 2D, výsledné sítě. . . . .	72
3.37	Úloha typu B ve 2D, průběh ukazatele REZ. . . . .	74
3.38	Úloha typu B ve 2D, výsledné sítě. . . . .	74
3.39	Bisekce naznačené hrany. . . . .	80
3.40	Odstranění naznačené hrany. . . . .	80
3.41	Přehození naznačené hrany. . . . .	81

3.42	Posun společného uzlu. . . . .	81
3.43	Přípustná oblast pro pohyb vrcholu $P_i$ . . . . .	82
3.44	Určení vzdálenosti hrany $b$ od vrcholu $P_i$ . . . . .	83
3.45	Průběh funkce $\Psi_{P_i}$ pro $\delta_{min} = 0.5$ , $a = 4$ . . . . .	86
3.46	Omezení pohybu nehraničního vrcholu. . . . .	88
3.47	Omezení pohybu hraničního vrcholu. . . . .	89
3.48	Porovnání lineárních rekonstrukcí pro úlohu A v 1D. . . . .	96
3.49	Průběh ukazatele L2E pro úlohu A v 1D. . . . .	97
3.50	Průběh ukazatele $Q_{norm}$ pro úlohu A v 1D. . . . .	97
3.51	Porovnání trajektorií uzlů sítě pro úlohu A v 1D. . . . .	97
3.52	Porovnání trajektorií uzlů sítě pro úlohu A v 1D. . . . .	98
3.53	Porovnání lineárních rekonstrukcí pro úlohu B v 1D. . . . .	100
3.54	Průběh L2E pro úlohu B v 1D. . . . .	100
3.55	Průběh $Q_{norm}$ pro úlohu B v 1D. . . . .	100
3.56	Porovnání trajektorií uzlů sítě pro úlohu B v 1D. . . . .	101
3.57	Porovnání trajektorií uzlů sítě pro úlohu B v 1D. . . . .	101
3.58	Průběh $Q_{norm}$ pro testované metody pro úlohu A ve 2D. . . . .	104
3.59	Porovnání výsledných sítí pro testované metody, úloha A ve 2D. . . . .	105
3.60	Průběh normy gradientu a normy Hessovy matice funkce $u_A$ . . . . .	106
3.61	Průběh $Q_{norm}$ pro testované metody, úloha B ve 2D. . . . .	108
3.62	Porovnání výsledných sítí pro testované metody, úloha B ve 2D. . . . .	109
3.63	Pohyb po nelineární hranici, oblast $\Omega$ a počáteční síť. . . . .	110
3.64	Průběh funkce $u_C$ na oblasti $\Omega$ , průběh $Q_{norm}$ . . . . .	111
3.65	Úloha s nelineární hranicí, porovnání výsledných sítí. . . . .	112
4.1	Lineární interpolace numerického řešení v 1D. . . . .	117
4.2	Ilustrace posunutí vrchlu $P$ na novou pozici $\tilde{P}$ . . . . .	119
4.3	Evoluce přesného řešení Burgersovy rovnice. . . . .	120
4.4	Porovnání numerického řešení pro Burgersovu rovnici. . . . .	123
4.5	Porovnání numerického řešení pro Sodovu trubici. . . . .	126
4.6	Porovnání spočtených řešení pro Sodovu trubici. . . . .	127
4.7	Porovnání spočtených řešení pro modifikovaný Sodův problém. . . . .	135
4.8	Porovnání spočtených řešení pro Sodovu trubici ve 2D. . . . .	139
4.9	Porovnání spočtených řešení pro Sodovu trubici ve 2D. . . . .	139
4.10	Porovnání spočtených řešení pro Sodovu trubici ve 2D. . . . .	140
4.11	Porovnání spočtených řešení pro Sodovu trubici ve 2D. . . . .	140
4.12	Detail spočtených řešení pro Sodovu trubici ve 2D. . . . .	141
4.13	Porovnání metod pro Sodovu trubici, použito $J = 20$ objemů. . . . .	154
4.14	Porovnání metod pro Sodovu trubici, použito $J = 40$ objemů. . . . .	155
4.15	Porovnání metod pro Sodovu trubici, použito $J = 80$ objemů. . . . .	155
4.16	Porovnání metod pro Sodovu trubici, použito $J = 160$ objemů. . . . .	156
4.17	Porovnání metod pro Sodovu trubici, použito $J = 320$ objemů. . . . .	156

# Použité značení a seznam zkratek

V celé práci použito následující značení:

$N$  - prostorová dimenze problému

$Z^+$  - množina všech nezáporných celých čísel

$\delta_{ij}$  - Kroneckerovo delta,  $\delta_{ij} = 1$  když  $i = j$ , jinak  $\delta_{ij} = 0$

$\Omega_t \subset \mathbb{R}^3$  - oblast vyplněná tekutinou v čase  $t$ , pro oblast nezávislou na čase  $\Omega = \Omega_t$

$\partial\Omega$  - hranice oblasti  $\Omega$

$(0, T)$  - časový interval

$\mathcal{M} = \{(x, t); x \in \Omega_t, t \in (0, T)\} \subset \mathbb{R}^4$  - oblast na níž jsou definovány veličiny popisující proudění

$Q_T = \Omega \times (0, T)$  - časo-prostorový válec

$\rho = \rho(x, t)$  - hustota

$\mathbf{v} = \mathbf{v}(x, t)$  - vektor rychlosti,  $\mathbf{v} = (v_1, v_2, v_3)$

$p = p(x, t)$  - tlak

$\theta = \theta(x, t)$  - absolutní teplota

$E = E(x, t)$  - celková energie

$e$  - hustota vnitřní energie

$\mathcal{T} = \begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix}$  - tenzor napětí,  $\tau_{ij} = \tau_{ij}(x, t)$

$\mathbb{D}(\mathbf{v})$  - tenzor rychlosti deformace

$\mathbf{f} = \mathbf{f}(x, t)$  - vektor hustoty vnější objemové síly

$q = q(x, t)$  - hustota zdrojů tepla v tekutině

$c_p, c_v$  - specifická tepla při konstantním tlaku a při konstantním objemu

$\lambda, \mu$  - koeficienty viskozity

$k$  - konstanta tepelné vodivosti

$R > 0$  - plynová konstanta

$\gamma > 1$  - Poissonova adiabatická konstanta

$M$  - Machovo číslo

$a$  - rychlost zvuku:

$\mathbf{f}_s$  - Eulerovy nevazké toky

$\mathbf{R}_s$  - vazké toky

$\mathbf{w}$  - stavový vektor

$\mathbb{A}_s(\mathbf{w})$  - Jacobiho matice zobrazení  $\mathbf{f}_s$

$\mathcal{P}(\mathbf{w}, \mathbf{n})$  - tok veličiny  $\mathbf{w}$  ve směru  $\mathbf{n}$

$\mathbb{P}(\mathbf{w}, \mathbf{n})$  - Jacobiho matice toku  $\mathcal{P}(\mathbf{w}, \mathbf{n})$

$\lambda_j$  - vlastní čísla  $\mathbb{P}(\mathbf{w}, \mathbf{n})$

$\Omega_h$  - polygonální aproximace  $\Omega$

$\mathcal{D}_h$  - síť konečných objemů

$J$  - počet konečných objemů sítě  $\mathcal{D}_h$

$P_i$  -  $i$ -tý vrchol sítě

$e$  - hrana sítě

$D_i$  -  $i$ -tý konečný objem

$|D_i|$  -  $N$ -dimenzionální míra konečného objemu

$\Gamma_{ij}$  - společná hranice objemů  $D_i$  a  $D_j$

$|\Gamma_{ij}|$  -  $N - 1$  dimenzionální míra  $\Gamma_{ij}$

$x_i$  - uzlové body 1D sítě

$x_{i+\frac{1}{2}}$  - střed  $i$ -tého konečného objemu v 1D

$h_i$  - velikost i-tého konečného objemu v 1D  
 $\mathcal{T}_h$  - trojúhelníková síť konečných objemů  
 $\mathbf{w}_i^k$  - přibližné (konstantní) řešení na objemu  $D_i$  v čase  $t_k$   
 $\mathbf{w}_h^k$  - přibližné řešení metody konečných objemů na síti  $\mathcal{D}_h$  v čase  $t_k$   
 $\hat{\mathbf{w}}_i^k$  - polynomiální rekonstrukce řešení na objemu  $D_i$  v čase  $t_k$   
 $\mathbf{H}$  - numerický tok  
 $\tau_k$  - velikost k-tého časového kroku

$\Omega_c$  - logická výpočetní oblast  
 $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)$  - souřadnice v  $\Omega_c$   
 $\mathcal{T}_h^c$  - triangulace oblasti  $\Omega_c$   
 $\mathcal{L}_{\mathcal{T}_h^c}$  - prostor všech po částech lineárních polynomů na triangulaci  $\mathcal{T}_h^c$  v  $\Omega_c$   
 $\mathcal{L}_{\mathcal{T}_h^c} = [\mathcal{L}_{\mathcal{T}_h^c}]^2$   
 $E(\boldsymbol{\xi})$  - kvadratický funkcionál  
 $\mathbb{G}_k, \mathbb{G}$  - monitorovací funkce ve 2D  
 $G$  - monitorovací funkce v 1D  
 $g - g = \det(\mathbb{G})$   
 $u$  - skalární funkce  
 $w(\boldsymbol{x})$  - váhová funkce pro Winslowovu monitorovací funkci  
 $\tau_\nu$  -  $\nu$ -tý krok umělého času  $\tau$   
 $\Delta\tau$  - velikost časového kroku u metody umělého času  
 $\mathcal{J}$  - Jacobián transformace souřadnic  
 $\mathcal{T}$  - parametr rychlosti posunu sítě

$\mathbf{w}_h^*$  - lineární rekonstrukce získaná z  $\mathbf{w}_h^k$   
 $e_h$  - chyba diskretizace  
 $\pi_h$  - operátor nejlepší aproximace  
 $r_h$  - interpolační operátor  
 $\mathbb{H}$  - Hessova matice  
 $\mathbb{M}$  - modifikace Hessovy matice  
 $I_u$  - chyba interpolace  
 $\|e\|_u$  - Riemannova norma hrany sítě  
 $Q_{\mathcal{T}_h}$  - parametr kvality sítě

$Q_{P_i}$  - parametr kvality vrcholu  $P_i$   
 $\mathcal{K}_{P_i}$  - sjednocení všech objemů  $D_j$  obsahujících vrchol  $P_i$   
 $\mathcal{K}_{P_i}^{ad}$  - přípustná oblast pro pohyb vrcholu  $P_i$   
 $\Phi_{P_i}$  - minimalizovaná kriteriální funkce příslušející vrcholu  $P_i$   
 $B_{P_i}$  - bariérová funkce příslušející vrcholu  $P_i$   
 $\Psi_{P_i}$  - funkce příslušející vrcholu  $P_i$ , jež zajišťuje kontrolu úhlu

$h_{min}$  - délka nejkratší hrany sítě (2D), velikost nejmenšího objemu sítě (1D)  
 $h_{max}$  - délka nejdélší hrany sítě (2D), velikost největšího objemu sítě (1D)  
 $h_{avg}$  - průměrná velikost konečného objemu sítě v 1D  
 $\varepsilon$  - poměr mezi nejdélší a nejkratší hranou sítě  $\mathcal{D}_h$   
 $L1E_\rho, L2E_\rho$  - použité ukazatele přesnosti výpočtu  
 $T$  - doba potřebná pro běh výpočtu  
 $ITER$  - počet proběhlých iterací

ADER - Arbitrary Derivative  
 AMA - Anisotropic Mesh Adaptation

ALLMF - Arc-Length Like Monitor Function  
 BFGS - Broyden, Fletcher, Goldfarb, Shannoa  
 CFL - Courant-Friedrichs-Lewy  
 DRP - Derivative Riemann Problem  
 GMCL - Geometric Mass Conservation Law  
 GMRES - Generalized Minimal Residual Method  
 GRP - Generalized Riemann Problem  
 ENO - Essentially Non Oscillatory  
 HLL - Harten Lax Van Leer  
 HLLC - Harten Lax Van Leer Contact  
 MKO - Metoda Konečných Objemů  
 MMPDE - Moving Mesh Partial Diferential Equation  
 MUSCL - Monotone Upstream-Centered Scheme for Conservation Laws)  
 SOR - Successive Overrelaxation  
 VQPMM(M) - Vertex Quality Parameter Moving Mesh (Methods)  
 WENO - Weighted Essentially Non Oscillatory

Názvy metod adaptace sítě v tabulce 4.1 jsou následující:

(P) - prefix označující použití predikce  
 AT - Artificial Time, použit umělý čas pro pohyb uzlů  
 GS - použita Gaussova-Seidelova iterace pro pohyb uzlů  
 JAC - použita Jacobiova iterace pro pohyb uzlů  
 VQPMM - použit přístup metody VQPMM  
 MOVE - použit přístup MMPDE  
 PM - sufix označující použití perturbační metody pro přepočet řešení  
 WA - sufix označující použití váženého průměru pro přepočet řešení

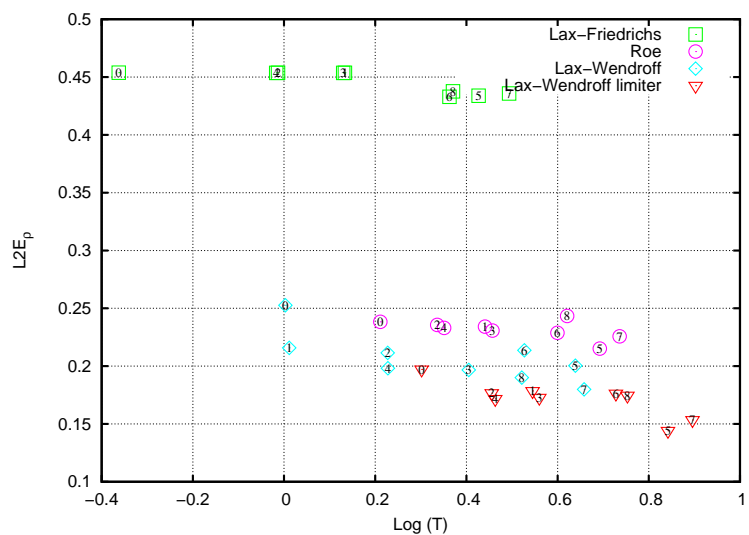
# Přílohová část

## Použitá konfigurace počítače

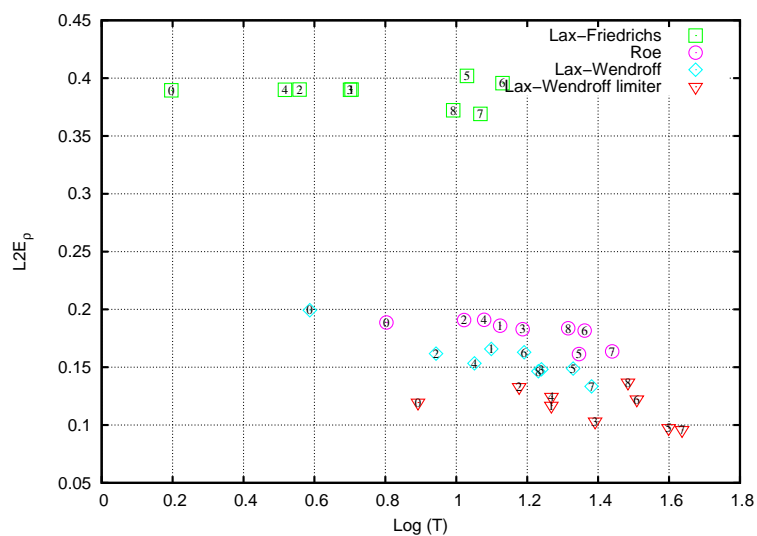
V naší práci jsme pro všechny numerické experimenty použili počítač, jehož konfigurace je následující:

- CPU: 4 Intel(R) Xeon(TM) CPU 3.20 GHz
- RAM: 2 GB
- Operační systém: Linux Debian 2.6.26-2-amd64

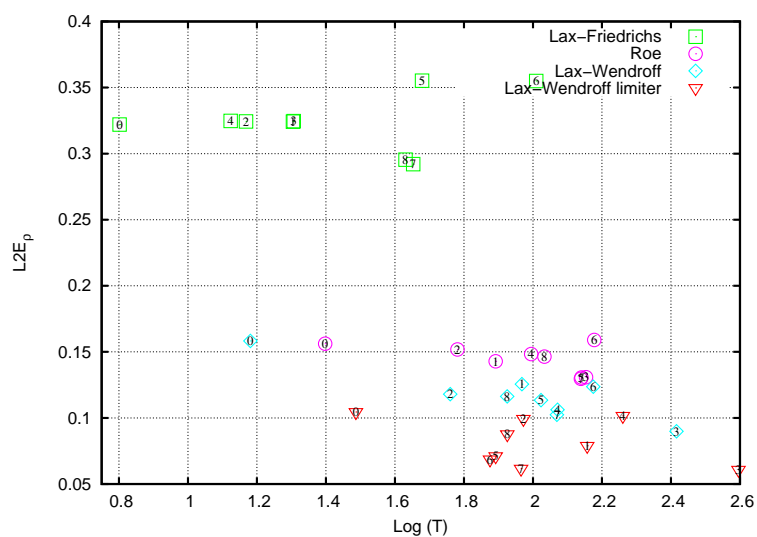
## Porovnání metod pro úlohu Sodovy trubice



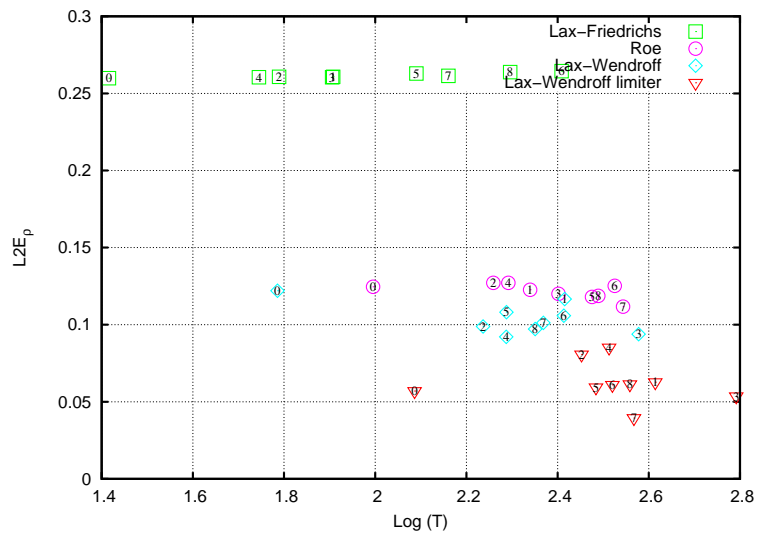
Obrázek 4.13: Porovnání metod pro  $J = 20$  objemů. Metody jsou očíslovány dle pořadí v tabulce 4.1, indexem 0 značíme výsledky získané bez adaptace.



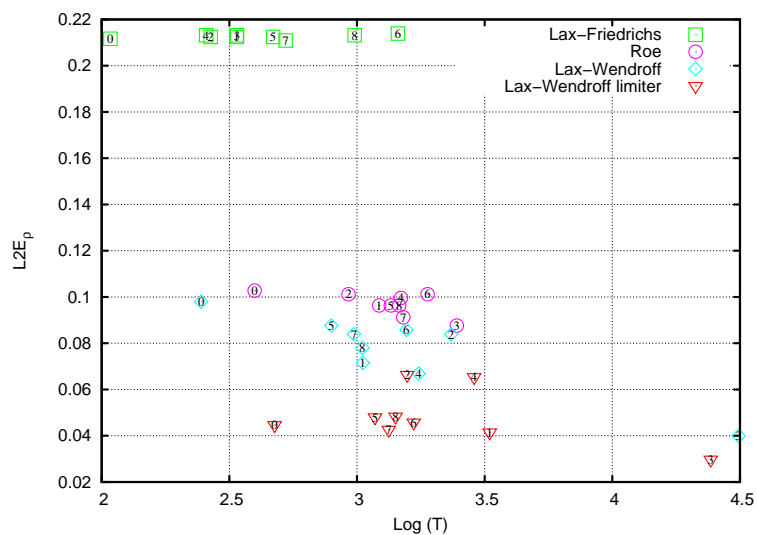
Obrázek 4.14: Porovnání metod pro  $J = 40$  objemů. Metody jsou očíslovány dle pořadí v tabulce 4.1, indexem 0 značíme výsledky získané bez adaptace.



Obrázek 4.15: Porovnání metod pro  $J = 80$  objemů. Metody jsou očíslovány dle pořadí v tabulce 4.1, indexem 0 značíme výsledky získané bez adaptace.



Obrázek 4.16: Porovnání metod pro  $J = 160$  objemů. Metody jsou očíslovány dle pořadí v tabulce 4.1, indexem 0 značíme výsledky získané bez adaptace.



Obrázek 4.17: Porovnání metod pro  $J = 320$  objemů. Metody jsou očíslovány dle pořadí v tabulce 4.1, indexem 0 značíme výsledky získané bez adaptace.