Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Libor Kadrnka

Schémata typu Moving Mesh pro řešení nestacionárních úloh

Katedra numerické matematiky

Vedoucí bakalářské práce: Doc. RNDr. Jiří Felcman, CSc.
Studijní program: Matematika, obecná matematika

2007

Chtěl bych poděkovat vedoucímu své bakalářské práce panu doc. RNDr. Jiřímu Felcmanovi, CSc., za cenné rady a připomínky během vedení mé bakalářské práce, panu Mgr. Petru Kuberovi za pomoc při programování, paní RNDr. Ivetě Hnětynkové, Ph.D., za zapůjčení počítače a zejména svým rodičům za podporu během celého mého studia.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 1. srpna 2007 Libor Kadrnka

Obsah

Název práce: Schémata typu Moving Mesh pro řešení nestacionárních úloh
Autor: Libor Kadrnka
Katedra (ústav): Katedra numerické matematiky
Vedoucí bakalářské práce: Doc. RNDr. Jiří Felcman, CSc.
e-mail vedoucího: felcman@karlin.mff.cuni.cz

Abstrakt: V předložené práci se zabýváme vytvořením adaptivní výpočetní sítě, která se používá při řešení parciálních diferenciálních rovnic (PDR). Síť budeme adaptovat v závislosti na chování přibližného řešení této rovnice v čase. Problém adaptace výpočetní sítě převedeme na minimalizaci jistého funkcionálu. Nutnou podmínkou k tomu, aby funkcionál nabýval minima, je splnění jisté diferenciální rovnice. Hledané uzly adaptované výpočetní sítě jsou tvořeny hodnotami diskrétního řešení této rovnice. Na nespojitých funkcích simulujících vývoj přibližného řešení PDR v čase ukážeme, jak adaptace sítě funguje. Poté s pomocí adaptivní výpočetní sítě najdeme přibližné řešení dvou PDR a toto řešení srovnáme s řešením přesným a s řešením získaným bez využití adaptivní sítě. Na závěr srovnáme metodu popsanou v této práci s anizotropní adaptivní metodou vyvíjenou na katedře numerické matematiky MFF UK.

Klíčová slova: adaptivní síť, geometrický zákon zachování, metoda konečných objemů, pohybující se nespojitost

Title: Moving Mesh schemes for nonstationary problems
Author: Libor Kadrnka
Department: Department of Numerical Mathematics
Supervisor: Doc. RNDr. Jiří Felcman, CSc.
Supervisor's e-mail address: felcman@karlin.mff.cuni.cz

Abstract: The presented work focuses on creating an adaptive computational mesh which is to be used for solving partial differential equations (PDE). The mesh is adapted according to the numerical solution behaviour of this equation in time. The problem of mesh adaptation is reduced to minimization of a certain functional. A necessary condition for the functional to have a minimum is that a certain differential equation holds. The grid points of the moving mesh on the next time level are given by the discrete solution of this equation. On discontinuous functions simulating the evolution of the PDE numerical solution in time we demonstrate how the mesh adaptation works. Then, using a moving mesh, we find a numerical solution to two PDEs and we compare this solution with the exact solution and with the solution obtained without using any adaptive mesh. Finally, we compare the adaptive method described in this work with an anisotropic mesh adaptation method developed at the Department of Numerical Mathematics MFF UK.

Keywords: adaptive mesh, geometric conservation law, finite volume method, moving discontinuity

Kapitola 1

Úvod

Funkce, které jsou řešením parciálních diferenciálních rovnic, obecně mohou obsahovat nespojitosti nebo mohou mít v některých bodech velkou hodnotu gradientu. Při hledání řešení numerickou metodou, jako je například metoda konečných objemů, může při použití rovnoměrné triangulace oblasti, na níž rovnici řešíme, dojít ke zkreslení tohoto řešení v místech, kde se tato nespojitost či vysoký gradient vyskytuje. Proto se používají tzv. adaptivní metody, kdy se v průběhu výpočtu triangulace oblasti v každém kroku přizpůsobuje přibližnému řešení získanému v kroku předchozím na základě daného kritéria adaptivní metody. V našem případě se v místech s vysokým gradientem nebo v místech nespojitosti zvýší počet prvků triangulace, zatímco na zbytku oblasti se jejich počet sníží, žádné nové prvky triangulace tedy nevytváříme, pouze se na základě chování získaného přibližného řešení na výpočetní oblasti snažíme prvky triangulace rozložit na výpočetní oblasti výše uvedeným způsobem.

Předpokládáme, že přibližné řešení parciální diferenciální rovnice se hledá *metodou konečných objemů*. Při použití této metody je přibližné řešení získáno ve tvaru po částech konstantní funkce, přičemž tato funkce je konstantní ve vnitřku každého prvku triangulace.

V této práci se budeme zabývat adaptivní metodou popsanou v článku [3], který se týká řešení jedno- a dvourozměrných hyperbolických rovnic, my se však omezíme pouze na jednorozměrný případ parciální diferenciální rovnice. Nejprve si tuto metodu podrobně představíme a následně si pomocí numerických experimentů ukážeme, jak funguje. Během experimentů nejdříve vyzkoušíme adaptivitu sítě na funkcích, u nichž známe jejich průběh, konkrétně budeme u funkcí, které jsou po částech lineární a jejichž nespojitost se v průběhu času pohybuje, zkoumat, zda se uzly sítě taktéž pohybují spolu s nespojitostí a zda se současně v daném časovém okamžiku v místě nespojitosti nalézá více uzlů než v místech, kde takovou vlastnost funkce nemá a vše si ilustrujeme na obrázcích. Poté použijeme adaptivní síť při řešení parciálních diferenciálních rovnic, je-

jichž přesné řešení známe a ukážeme, že adaptivita sítě výrazně zvyšuje přesnost přibližného řešení oproti případu, kdy se síť neadaptuje. Vše bude opět ilustrováno na obrázcích.

Na závěr srovnáme zkoumanou adaptivní metodu s jinou adaptivní metodou, a to anizotropní metodou zjemňování výpočetní sítě popsanou v článku [2] a na internetové adrese http://www.karlin.mff.cuni.cz/~dolejsi/angen/angen3.1.htm a na obrázcích srovnáme pohyb uzlů výpočetní sítě při použití obou adaptivních metod.

Kapitola 2

Popis metody

V této kapitole si podrobně představíme adaptivní metodu popsanou v článku [3], veškeré vzorce v této kapitole jsou citovány z [3].

2.1 Vytvoření sítě pomocí minimalizace funkcionálu

Předpokládejme, že $N \geq 1$ označuje dimenzi prostoru, v němž se nachází *fyzikální oblast* $\Omega_p \subset \mathbb{R}^N$, tj. oblast, v níž máme definovanou parciální diferenciální rovnici, při jejímž řešení budeme adaptovat výpočetní síť. Nechť souřadnice fyzikální oblasti jsou dány jako $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, předpokládejme dále, že fyzikální oblast je obrazem *výpočetní oblasti* $\Omega_c \subset \mathbb{R}^N$ při vzájemně jednoznačném zobrazení z Ω_c na Ω_p, označme $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)$ souřadnice této výpočetní oblasti. Transformaci souřadnic z oblasti Ω_c do oblasti Ω_p označíme

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in \Omega_c,$$

inverzní zobrazení z oblasti Ω_p do oblasti Ω_c označíme

$$\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega_p.$$

Při hledání řešení zmiňované parciální diferenciální rovnice provedeme triangulaci fyzikální oblasti Ω_p a následně hledáme přibližné řešení. Přesné řešení může mít v některých místech velký gradient či dokonce nespojitost a není-li triangulace oblasti dostatečně jemná, nemusí mít numerické řešení námi požadovanou přesnost. Na druhou stranu, je-li triangulace oblasti rovnoměrná a má-li být dostatečně jemná, můžeme tímto požadavkem na jemnost sítě značně zvýšit výpočetní náročnost, což je nežádoucí. Proto je vhodné triangulaci zjemňovat pouze v těch místech, ve kterých dochází k velkým změnám v řešení. Naším cílem je tedy nalézt takové zobrazení z pomocné výpočetní oblasti Ω_c do oblasti

Ω_p, tj. takové rozložení prvků triangulace, aby se v takto problematických místech nacházel větší počet prvků triangulace nežli v místech, kde výše zmíněné problémy nenastávají.

Toho lze dosáhnout například minimalizací následujícího funkcionálu, jak je uvedeno v článku [3]. Minimalizujme tedy funkcionál

$$E(\boldsymbol{\xi}) = \frac{1}{2}\sum_{k=1}^{N}\int_{\Omega_p} \nabla\xi_k^T G_k^{-1}\nabla\xi_k d\boldsymbol{x}, \tag{2.1}$$

kde $\nabla := (\partial_{x_1}, \partial_{x_2}, \ldots, \partial_{x_N})^T$ a $G_k,\ 1 \leq k \leq N$ jsou dané symetrické pozitivně definitní matice zvané *monitorovací funkce*.

Nutnou podmínkou, aby funkcionál E nabýval minima, je *Euler-Lagrangeova rovnice* funkcionálu (2.1), a to

$$\nabla.(G_k^{-1}\nabla\xi_k) = 0, \qquad 1 \leq k \leq N. \tag{2.2}$$

Za monitorovací funkce G_k zvolíme $G_k = \omega I,\ 1 \leq k \leq N$, kde I je jednotková matice a ω je kladná váhová funkce, např. $\omega = \sqrt{1 + |\nabla u|^2}$, kde u je řešením parciální diferenciální rovnice, které hledáme. Rovnost (2.2) pak lze zapsat ve tvaru

$$\nabla.\left(\frac{1}{\omega}\nabla\xi_k\right) = 0, \qquad 1 \leq k \leq N. \tag{2.3}$$

2.1.1 Jednorozměrný případ

Zabývejme se nyní případem, kdy $N = 1$. Nechť x označuje fyzikální souřadnice definované v odstavci 2.1. Nechť $x \in \Omega_p = [a, b],\ a,\ b \in \mathbb{R},\ a < b$. Triangulace intervalu $[a, b]$ je v tomto jednoduchém případě dělení intervalu.

V této práci se budeme zabývat adaptivním zjemňováním výpočetní sítě na intervalu $[a, b]$ při hledání řešení parciální diferenciální rovnice tvaru

$$\frac{\partial u}{\partial t}(x,t) + \frac{\partial\,(f(u))}{\partial x}(x,t) = 0, \quad t \in (0,T], \quad x \in [a,b], \tag{2.4}$$

s počáteční podmínkou

$$u(x,0) = u_0(x), \quad x \in [a,b], \tag{2.5}$$

a vhodnou okrajovou podmínkou.
Předpokládáme, že $f \in C^1(\mathbb{R})$ a že funkce u_0, která má kompaktní nosič

v $\mathbb{R}$, splňuje $u_0 \in L^\infty(\mathbb{R}) \cap BV(\mathbb{R})$. O přesném řešení u této rovnice budeme předpokládat, že $u \in L^\infty(\mathbb{R})$ a že u je slabým po částech hladkým řešením rovnice (2.4) (definice pojmu *slabé po částech hladké řešení* viz kapitola 2.3 v knize [1]).

Definujme si pomocnou výpočetní oblast $\Omega_c = [0,1]$, označme $\xi \in [0,1]$ souřadnice na $[0,1]$ a definujme také transformace souřadnic

$$\begin{aligned} x &= x(\xi), \qquad \xi \in [0,1], \\ x(0) &= a, \qquad x(1) = b. \end{aligned} \tag{2.6}$$

a

$$\begin{aligned} \xi &= \xi(x), \qquad x \in [a,b], \\ \xi(a) &= 0, \qquad \xi(b) = 1. \end{aligned} \tag{2.7}$$

Nyní najdeme zobrazení z $[0,1]$ na $[a,b]$ takové, abychom pomocí něj zkonstruovali adaptivní výpočetní síť na $[a,b]$. To provedeme pomocí minimalizace funkcionálu (2.1), konkrétně podmínky (2.3), která má v případě $N = 1$ tvar

$$(\omega^{-1}\xi_x)_x = 0 \text{ na } [a,b], \tag{2.8}$$

kde monitorovací funkci ω definujeme jako

$$\omega(x) = \sqrt{1 + \left(\frac{\partial u}{\partial x}(x)\right)^2}, \quad x \in [a,b], \tag{2.9}$$

kde u je přesné řešení rovnice (2.4).

Výraz na levé straně rovnice (2.8) je funkce proměnné x. Kdyby se nám podařilo tuto rovnici transformovat do takového tvaru, aby levá strana transformované rovnice byla funkcí proměnné ξ, diskrétní přibližné řešení x transformované rovnice nám následně umožní, jak dále uvidíme, zkonstruovat uzly adaptivního dělení intervalu $[a,b]$. Takto získané diskrétní řešení bude totiž minimalizovat funkcionál (2.1), protože rovnice (2.8) je pouze jednorozměrnou podmínkou (2.2).

Transformujme tedy rovnici (2.8). K tomu využijeme následující postup.

Nechť $f : X \to \mathbb{R}$, $g : Y \to X$, $g^{-1} : X \to Y$, $f \in C^1(X)$, $g \in C^1(Y)$, $g^{-1} \in C^1(X)$, kde X, Y jsou uzavřené intervaly v $\mathbb{R}$ a g je transformace souřadnic z Y na X. Platí tedy $x = g(\xi)$, $\xi = g^{-1}(x)$ $\forall x \in X$, $\forall \xi \in Y$, $\frac{dg}{d\xi} \neq 0$ v Y a $\frac{dg^{-1}}{dx} \neq 0$ v X. Rovněž platí $\forall x \in X$, $\forall \xi \in Y$

$$\frac{d}{d\xi}(f \circ g)(\xi) \;=\; \frac{df}{dx}(g(\xi)) \,.\, \frac{dg}{d\xi}(\xi) = \frac{df}{dx}(x) \,.\, \frac{dg}{d\xi}(g^{-1}(x)) =$$

$$= \frac{df}{dx}(x) \cdot \frac{1}{\frac{dg^{-1}}{dx}(x)},$$

kde poslední člen $\frac{dg}{d\xi}(g^{-1}(x)) = \frac{1}{\frac{dg^{-1}}{dx}(x)} \neq 0 \ \forall x \in X$, a proto platí následující ekvivalence:

$$\forall \xi \in Y: \ \frac{d}{d\xi}(f \circ g)(\xi) = 0 \qquad \Leftrightarrow \qquad \forall x \in X: \ \frac{df}{dx}(x) = 0.$$

Aplikací předchozího postupu na funkce

$$g(\xi) := x(\xi), \quad \xi \in [0,1],$$

$$g^{-1}(x) := \xi(x), \quad f(x) := \frac{1}{\omega(x)} \cdot \frac{dg^{-1}}{dx}(x), \qquad x \in [a,b]$$

a s využitím následujícího faktu, při jehož odvození se použije věta o derivaci inverzní funkce:

$$\frac{\xi_x}{\omega} = \frac{\frac{d\xi}{dx}(x)}{\omega(x)} = \frac{1}{\omega(x(\xi))} \cdot \frac{1}{\frac{dx}{d\xi}(\xi)} = \frac{1}{\widetilde{\omega}(\xi)} \cdot \frac{1}{\frac{dx}{d\xi}(\xi)} = \frac{1}{\widetilde{\omega} x_\xi},$$

$$\forall \xi \in [0,1] \ \forall x \in [a,b],$$

kde

$$\widetilde{\omega}(\xi) = \omega(x(\xi)), \qquad \xi \in [0,1].$$

Dostáváme pro $x = x(\xi), \ \xi = \xi(x), \ x \in [a,b], \ \xi \in [0,1]$:

$$0 = (\omega^{-1}\xi_x)_x \qquad \Leftrightarrow \qquad 0 = \left(\frac{1}{\widetilde{\omega} x_\xi}\right)_\xi = \frac{-\widetilde{\omega}_\xi x_\xi - \widetilde{\omega} x_{\xi\xi}}{(\widetilde{\omega} x_\xi)^2} \qquad \Leftrightarrow$$

$$\Leftrightarrow \qquad \widetilde{\omega}_\xi x_\xi + \widetilde{\omega} x_{\xi\xi} = 0 \qquad \Leftrightarrow \qquad (\widetilde{\omega} x_\xi)_\xi = 0.$$

Vidíme, že rovnost (2.8) je ekvivalentní rovnosti

$$(\widetilde{\omega} x_\xi)_\xi = 0 \text{ na } [0,1], \tag{2.10}$$

čímž dostáváme požadovanou transformaci rovnice (2.8), kde výraz na levé straně závisel na proměnné x, na rovnici, kde je již výraz na levé straně funkcí proměnné ξ. Konstrukci uzlů adaptivní sítě na intervalu $[a,b]$ provedeme v následujícím odstavci.

2.2 Popis algoritmu

Na základě odstavce 2.1.1 již nyní můžeme zkonstruovat uzly adaptivní výpočetní sítě v jednorozměrném případě parciální diferenciální rovnice (2.4). Toho dosáhneme nalezením přibližného řešení rovnice

$$(\widetilde{\omega} x_\xi)_\xi = 0 \text{ na } [0,1] \tag{2.11}$$

s okrajovými podmínkami

$$x(0) = a, \quad x(1) = b. \tag{2.12}$$

Ze získaného diskrétního řešení už můžeme jednoduše zkonstruovat uzly adaptivní sítě, jež budou splňovat naše požadavky, to znamená že v místech, kde bude mít funkce u velký gradient nebo nespojitost, bude vyšší počet uzlů dělení intervalu $[a, b]$, na němž je funkce definována, než v místech, kde funkce u takovou vlastnost nemá.

Zaveďme označení. Uvažujme interval $[0, 1]$, nechť $\xi \in [0, 1]$ a buď $n \in \mathbb{N}$, $n \geq 2$. Nechť $h = \frac{1}{n-1}$ je krok rovnoměrného dělení intervalu $[0, 1]$ a $\xi_j = (j-1)h$, $1 \leq j \leq n$ jsou uzly sítě. Označme $0 = t_0 < t_1 < \ldots < t_l = T$, $l \in \mathbb{N}$ dělení intervalu $[0, T]$. V každém časovém kroku t_k, $k \in \{0, \ldots, l\}$, hledání přibližného řešení rovnice (2.4) označme $u^k(x) := u(x, t_k)$, $x \in [a, b]$. Protože ke zjemňování výpočetní sítě dochází pouze na základě přibližných hodnot funkce u získaných v daném časovém kroku t_k, a zjemňování tedy nezávisí na předchozích hodnotách funkce u v časech $t_1, \ldots, t_{k-1}$, bez nebezpečí nedorozumění budeme dále index k u funkce u^k vynechávat.

2.2.1 Metoda sítí

Nejdříve zkusíme nalézt řešení rovnice (2.11) *metodou sítí*. Přibližné řešení budeme hledat ve vnitřních bodech dělení intervalu [0,1] takových, aby všechny aproximace byly dobře definovány.

Pro $1 \leq j \leq n$ označme $x(\xi_j) \approx x_j$ aproximaci hodnot funkce x v bodě ξ_j a pro $1 \leq j \leq n-1$ $x(\xi_{j+\frac{1}{2}}) \approx x_{j+\frac{1}{2}}$ aproximaci hodnot funkce x v bodech $\xi_{j+\frac{1}{2}} = \frac{\xi_j + \xi_{j+1}}{2}$, $1 \leq j \leq n-1$. Dále pro $1 \leq j \leq n-1$ označme $u(\widehat{x}_{j+\frac{1}{2}}) \approx u_{j+\frac{1}{2}}$ aproximaci hodnot funkce u v bodě $\widehat{x}_{j+\frac{1}{2}} = \frac{x_j + x_{j+1}}{2}$, $1 \leq j \leq n-1$.

Aproximujme derivace rovnice (2.11) pomocí centrálních diferencí:

$$x_\xi(\xi_j) = \frac{dx}{d\xi}(\xi_j) \approx \frac{x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}}{h}, \qquad 2 \leq j \leq n-1.$$

Dále aproximujme druhou derivaci centrálními diferencemi za pomoci

předchozí aproximace:

$$(\widetilde{\omega}x_\xi)_\xi(\xi_j) \approx \frac{\widetilde{\omega}(\xi_{j+\frac{1}{2}})(x_{j+1}-x_j) - \widetilde{\omega}(\xi_{j-\frac{1}{2}})(x_j - x_{j-1})}{h^2},$$

$$2 \le j \le n-1.$$

Zbývá aproximovat hodnoty $\widetilde{\omega}(\xi_{j+\frac{1}{2}}) = \omega(x_{j+\frac{1}{2}}),\ 2 \le j \le n-1$. Zkusme to opět pomocí centrálních diferencí takto, udělejme to pouze pro $3 \le j \le n-2$:

$$\widetilde{\omega}(\xi_{j+\frac{1}{2}}) = \omega(x_{j+\frac{1}{2}}) \approx \sqrt{1 + \left(\frac{u_{j+\frac{3}{2}} - u_{j-\frac{1}{2}}}{\widehat{x}_{j+\frac{3}{2}} - \widehat{x}_{j-\frac{1}{2}}}\right)^2} =$$

$$= \sqrt{1 + 4\left(\frac{u_{j+\frac{3}{2}} - u_{j-\frac{1}{2}}}{x_{j+2} + x_{j+1} - x_j - x_{j-1}}\right)^2}, \qquad 3 \le j \le n-2.$$

Je vidět, že abychom mohli aproximovat $\omega(x_{j+\frac{1}{2}}),\ 3 \le j \le n-2$, musíme znát hodnoty $u_{j+\frac{1}{2}},\ 3 \le j \le n-2$, přičemž $u_{j+\frac{1}{2}} \approx u(\widehat{x}_{j+\frac{1}{2}}) = u(\frac{x_j + x_{j+1}}{2}),\ 3 \le j \le n-2$, tedy koeficienty $u_{j+\frac{1}{2}},\ 3 \le j \le n-2$ síťové rovnice

$$\frac{\sqrt{1 + 4\left(\frac{u_{j+\frac{3}{2}} - u_{j-\frac{1}{2}}}{x_{j+2} + x_{j+1} - x_j - x_{j-1}}\right)^2}\,(x_{j+1} - x_j) - \sqrt{1 + 4\left(\frac{u_{j+\frac{1}{2}} - u_{j-\frac{3}{2}}}{x_{j+1} + x_j - x_{j-1} - x_{j-2}}\right)^2}\,(x_j - x_{j-1})}{h^2} = 0,$$

$$3 \le j \le n-2,$$

závisí na síťových uzlech $x_j,\ x_{j+1}$, které však počítáme, tedy hodnoty $u_{j+\frac{1}{2}},\ 3 \le j \le n-2$ předem neznáme. Protože z rovnice (2.9) dostáváme, že funkce ω závisí nejen na hodnotě proměnné x, ale také na hodnotě derivace funkce u proměnné x, a dále z konstrukce aproximace $\omega(x_{j+\frac{1}{2}})$, $3 \le j \le n-2$, je vidět, že síťová rovnice daná touto aproximací bude vždy obsahovat koeficienty, které budou záviset na uzlech sítě, tj. na řešení, které hledáme. To značně komplikuje výpočet, neboť dostáváme soustavu nelineárních rovnic a nalezení řešení takové soustavy je výpočetně mnohem náročnější než u soustavy lineární. Užití metody sítí na problém (2.11) tedy není vhodné, a proto použijeme následující, i když méně efektivní metodu, a to *metodu umělého času*.

2.2.2 Metoda umělého času

Metoda umělého času je založena na hledání stacionárního řešení úlohy, kterou získáme modifikací rovnice (2.11). Zaveďme funkci $x(\xi, \tau)$

proměnných ξ a τ, kde τ je tzv. *umělý čas*, bez nebezpečí nedorozumění ji budeme značit stejně jako funkci původní, a řešme rovnici

$$x_\tau = (\widetilde{\omega} x_\xi)_\xi \ , \quad \xi \in (0,1),\ \tau > 0 \tag{2.13}$$

s okrajovými podmínkami

$$x(0,\tau) = a,\ x(1,\tau) = b, \quad \tau > 0$$

a počáteční podmínkou

$$x(\xi,0) = (b-a)\xi + a, \quad \xi \in [0,1],$$

která odpovídá počátečnímu rovnoměrnému rozložení uzlů výpočetní sítě na $[a,b]$. Bude-li časová derivace funkce $x(\xi,\tau)$ nulová, bude schéma založené na metodě umělého času řešit rovnici (2.11).

Označme $\Delta\tau > 0$ velikost kroku diskretizace v proměnné τ, buď $\tau_\nu = \nu\ \Delta\tau$, $\nu \in I\!N_0$, krok této diskretizace, definujme $x(\xi_j,\tau_\nu) \approx x_j^{[\nu]}$, $1 \le j \le n$, $\nu \in I\!N_0$, aproximaci funkce $x(\xi,\tau)$ v bodě (ξ_j,τ_ν), $1 \le j \le n$, $\nu \in I\!N_0$. Pro $1 \le j \le n-1$ a $\nu \in I\!N_0$ označme $u(\widehat{x}_{j+\frac{1}{2}}^{[\nu]}) \approx u_{j+\frac{1}{2}}^{[\nu]}$ aproximaci hodnot funkce u v bodě $\widehat{x}_{j+\frac{1}{2}}^{[\nu]}$, kde $\widehat{x}_{j+\frac{1}{2}}^{[\nu]} = \frac{x_j^{[\nu]} + x_{j+1}^{[\nu]}}{2}$, $1 \le j \le n-1$, $\nu \in I\!N_0$. Funkci $\widetilde{\omega}(\xi_{j+\frac{1}{2}})$ z odstavce 2.2.1 nyní budeme značit jako $\omega(u_{j+\frac{1}{2}})$, $1 \le j \le n-1$, aby byla patrná závislost funkce ω na funkci u (ve skutečnosti ovšem funkce ω závisí na ∇u).

Diskretizujme rovnici (2.13), dostaneme:

$$\frac{x_j^{[\nu+1]} - x_j^{[\nu]}}{\Delta\tau} = \frac{\omega(u_{j+\frac{1}{2}}^{[\nu]})(x_{j+1}^{[\nu]} - x_j^{[\nu]}) - \omega(u_{j-\frac{1}{2}}^{[\nu]})(x_j^{[\nu]} - x_{j-1}^{[\nu]})}{h^2}\ , \tag{2.14}$$

$$2 \le j \le n-1, \quad \nu \in I\!N_0$$

z čehož dostáváme diferenční schéma

$$x_j^{[\nu+1]} = x_j^{[\nu]} + \frac{\Delta\tau}{h^2}[\omega(u_{j+\frac{1}{2}}^{[\nu]})(x_{j+1}^{[\nu]} - x_j^{[\nu]}) - \omega(u_{j-\frac{1}{2}}^{[\nu]})(x_j^{[\nu]} - x_{j-1}^{[\nu]})]\ , \tag{2.15}$$

$$2 \le j \le n-1, \quad \nu \in I\!N_0$$

s okrajovými podmínkami $x_0^{[\nu]} = a$ a $x_n^{[\nu]} = b\ \forall \nu \in I\!N_0$.

Funkci

$$\omega(u_{j+\frac{1}{2}}) = \sqrt{1 + \left(\frac{\partial u}{\partial x}(\widehat{x}_{j+\frac{1}{2}})\right)^2}\ , \quad 1 \le j \le n-1$$

aproximujeme například pomocí centrálních diferencí.

Nyní ukážeme, že schéma (2.15) řeší rovnici (2.11). Skutečně, jestliže $| x_j^{[\nu+1]} - x_j^{[\nu]} | \approx 0$, je pravá strana rovnice (2.14) také přibližně rovna nule a uzly $\{x_j^{[\nu+1]}\}$ splňují diferenční schéma pro přibližné řešení rovnice (2.11). V praxi se tato podmínka aplikuje tak, že při hledání stacionárního řešení rovnice (2.13) předepíšeme toleranci ε a provádíme iterace schématu (2.15) pro $\nu = 0,\ 1, \dots$, dokud není splněno $| x_j^{[\nu+1]} - x_j^{[\nu]} | < \varepsilon$.

Schéma (2.15) můžeme přepsat do tvaru

$$x_j^{[\nu+1]} = \alpha_{j+\frac{1}{2}} x_{j+1}^{[\nu]} + (1 - \alpha_{j+\frac{1}{2}} - \alpha_{j-\frac{1}{2}}) x_j^{[\nu]} + \alpha_{j-\frac{1}{2}} x_{j-1}^{[\nu]}, \tag{2.16}$$

$$2 \le j \le n-1,$$

kde $\nu \in \mathbb{N}_0$ je krok iterace a

$$\alpha_{j+\frac{1}{2}} = \frac{\Delta\tau}{h^2} \omega(u_{j+\frac{1}{2}}^{[\nu]}), \qquad 2 \le j \le n-1.$$

Uvažujeme následující podmínku stability

$$\max_{1 \le j \le n-1} \alpha_{j+\frac{1}{2}} \le \frac{1}{2}\ . \tag{2.17}$$

V praxi je ovšem schéma (2.16) příliš pomalé a nedává dobré výsledky. Místo schématu (2.16) proto budeme uvažovat schéma

$$x_j^{[\nu+1]} = \frac{\omega(u_{j+\frac{1}{2}}^{[\nu]}) x_{j+1}^{[\nu]} + \omega(u_{j-\frac{1}{2}}^{[\nu]}) x_{j-1}^{[\nu+1]}}{\omega(u_{j+\frac{1}{2}}^{[\nu]}) + \omega(u_{j-\frac{1}{2}}^{[\nu]})}\ , \quad 2 \le j \le n-1. \tag{2.18}$$

V praktických výpočtech je tato metoda mnohem rychlejší. Schéma (2.18) lze odvodit z předešlého schématu. Všimněme si, že ve vzorci (2.16) se hodnota uzlu $x_j^{[\nu+1]}$ získá pomocí jisté kombinace uzlů $x_{j+1}^{[\nu]}$, $x_j^{[\nu]}$ a $x_{j-1}^{[\nu]}$, tedy pro výpočet hodnoty uzlu v iteraci $\nu+1$ používáme hodnoty pouze z předchozího kroku ν. Pokusme se výpočet urychlit tím, že k výpočtu $x_j^{[\nu+1]}$ použijeme již známé hodnoty z kroku $\nu+1$. Dostaneme rovnici

$$x_j^{[\nu+1]} = \alpha_{j+\frac{1}{2}} x_{j+1}^{[\nu]} + (1 - \alpha_{j+\frac{1}{2}} - \alpha_{j-\frac{1}{2}}) x_j^{[\nu+1]} + \alpha_{j-\frac{1}{2}} x_{j-1}^{[\nu+1]},$$

$$2 \le j \le n-1,$$

z níž si vyjádříme $x_j^{[\nu+1]}$ a získáme rovnici (2.18).

V článku [3] se používá ještě tzv. „zhlazování“ monitorovací funkce ω ve tvaru

$$\omega(u_{j+\frac{1}{2}}) \leftarrow \frac{1}{4}[\omega(u_{j+\frac{3}{2}}) + 2\omega(u_{j+\frac{1}{2}}) + \omega(u_{j-\frac{1}{2}})], \qquad 2 \le j \le n-2.$$

Funkce ω má totiž takový průběh, že nabývá velkých hodnot v místech, ve kterých nabývá velkých hodnot gradient u a je-li gradient u malý, je funkce ω malá. V závislosti na hodnotách funkce ω se následně adaptuje síť, čím vyšší jsou tyto hodnoty, tím více uzlů se v daném místě nashromáždí. Problém nastává, je-li gradient u v jistém smyslu velký. V takovém případě nabývá funkce ω v daném místě extrémně vysokých hodnot a pro splnění podmínky stability (2.17) je nutné použít velmi malý krok umělého času $\Delta\tau$, což zpomaluje výpočet. Vhodné použití výše zmíněného zhlazování má za následek, že k nabývání takto extrémních hodnot u funkce ω nedochází a podmínka stability je splněna i pro vyšší hodnoty $\Delta\tau$, a tedy výpočet probíhá rychleji.

Při výpočtu nových hodnot uzlů potřebujeme obvykle vykonat několik iterací schématu (2.18), abychom získali dostatečně jemnou síť. V takovém případě však vzniká problém. Změníme-li výpočetní síť, neznáme hodnoty přibližného řešení, které jsou definované na předchozí síti, na síti nové. Jedním z řešení je hodnoty na původní síti přepočítat na hodnoty pro novou síť v každé iteraci zjemňování sítě tak, aby nové hodnoty splňovaly určité podmínky. My použijeme následující postup. Označme $u_{j+\frac{1}{2}}^{[\nu]}$ hodnotu $u_{j+\frac{1}{2}}$ v ν-té iteraci adaptivní metody, $1 \le j \le n-1$, $\nu \in \mathbb{N}_0$. Hodnotu $u_{j+\frac{1}{2}}^{[\nu+1]}$ spočteme jako

$$u_{j+\frac{1}{2}}^{[\nu+1]} = \beta_j^{[\nu]} u_{j+\frac{1}{2}}^{[\nu]} - \gamma_j^{[\nu]}((\widehat{cu})_{j+1}^{[\nu]} - (\widehat{cu})_j^{[\nu]}), \qquad 1 \le j \le n-1, \tag{2.19}$$

kde

$$\gamma_j^{[\nu]} = (x_{j+1}^{[\nu+1]} - x_j^{[\nu+1]})^{-1}\ , \qquad \beta_j^{[\nu]} = \gamma_j^{[\nu]}(x_{j+1}^{[\nu]} - x_j^{[\nu]}), \qquad 1 \le j \le n-1,$$

a *numerický tok* $(\widehat{cu})_j$ počítáme jako

$$(\widehat{cu})_j^{[\nu]} = \frac{c_j^{[\nu]}}{2}(u_j^+ + u_j^-) - \frac{|c_j^{[\nu]}|}{2}(u_j^+ - u_j^-), \qquad 2 \le j \le n-2.$$

Protože $\forall \nu \in \mathbb{N}_0$ platí $x_1 = a$ a $x_n = b$, položme

$$(\widehat{cu})_1^{[\nu]} := 0, \quad (\widehat{cu})_{n-1}^{[\nu]} := 0.$$

V definici numerického toku dále máme

$$c_j^{[\nu]} = x_j^{[\nu]} - x_j^{[\nu+1]}\ , \qquad 1 \le j \le n,$$

a

$$u_j^{\pm} = u_{j\pm\frac{1}{2}} + \frac{1}{2}(x_j - x_{j\pm1})\widetilde{S}_{j\pm\frac{1}{2}}\ , \quad 3 \le j \le n-2,$$

přičemž hodnoty $\widetilde{S}_{j+\frac{1}{2}}$ definujeme

$$\widetilde{S}_{j+\frac{1}{2}} = \left(\operatorname{sgn}(\widetilde{S}^{+}_{j+\frac{1}{2}}) + \operatorname{sgn}(\widetilde{S}^{-}_{j+\frac{1}{2}})\right) \frac{|\widetilde{S}^{+}_{j+\frac{1}{2}} \widetilde{S}^{-}_{j+\frac{1}{2}}|}{|\widetilde{S}^{+}_{j+\frac{1}{2}}| + |\widetilde{S}^{-}_{j+\frac{1}{2}}|}\ , \qquad 2 \le j \le n-2,$$

$$\widetilde{S}^{+}_{j+\frac{1}{2}} = \frac{u_{j+\frac{3}{2}} - u_{j+\frac{1}{2}}}{x_{j+\frac{3}{2}} - x_{j+\frac{1}{2}}}\ , \qquad 1 \le j \le n-2,$$

$$\widetilde{S}^{-}_{j+\frac{1}{2}} = \frac{u_{j+\frac{1}{2}} - u_{j-\frac{1}{2}}}{x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}}\ , \qquad 2 \le j \le n-1.$$

Hodnoty $u^{\pm}_{j}$ pro $j \in \{1,\ 2,\ n-1,\ n\}$ definujeme analogicky na základě okrajových podmínek řešené parciální diferenciální rovnice.
Jedním z důvodů použití této metody přepočítávání hodnot $u_{j+\frac{1}{2}}$ je, že splňuje *geometrický zákon zachování*, tj. platí

$$\sum_{j=1}^{n-1} (x^{[\nu+1]}_{j+1} - x^{[\nu+1]}_{j}) u^{[\nu+1]}_{j+\frac{1}{2}} = \sum_{j=1}^{n-1} (x^{[\nu]}_{j+1} - x^{[\nu]}_{j}) u^{[\nu]}_{j+\frac{1}{2}} \qquad \forall \nu \in I\!N_0.$$

Algoritmus řešení PDR s využitím adaptivní sítě:

0. Inicializace

Zvolme $n \in I\!N,\ n \ge 2$, počet uzlů fyzikální oblasti $\Omega_p = [a,b]$. Definujme rovnoměrné rozložení bodů $\xi_j = (j-1)h,\ \ 1 \le j \le n$, v oblasti $\Omega_c = [0,1]$, $h = \frac{1}{n-1}$. Definujme dělení intervalu $[0,T]$ jako $0 = t_0 < t_1 < t_2 < \ldots < t_l = T$ pro nějaké $l \in I\!N$.

Buď $\nu \in I\!N_0$ a $k \in I\!N_0$.

Nechť $\{x^k_j\}_{j=1}^{n}$ označuje uzly výpočetní sítě tvořící v reálném čase t_k konečné objemy $D^k_j = [x^k_j, x^k_{j+1}]$ na [a,b] pro $1 \le j \le n-1$.

Označme aproximaci $u(\ .\ , t_k)|_{D^k_j} \approx u^k_{j+\frac{1}{2}},\ 1 \le j \le n-1$, přesného řešení u rovnice (2.4) na konečném objemu $D^k_j = [x^k_j, x^k_{j+1}]$ v reálném čase t_k.

Nechť $x^{k[\nu]}_j$, $1 \le j \le n$, jsou uzly výpočetní sítě v ν-tém kroku adaptace sítě (2.18) - (2.19), kterou provádíme v reálném čase t_k, příslušné počáteční rozložení uzlů je $x^{k[0]}_j := x^k_j$. Pro jednoduchost budeme index k vynechávat, tj. místo $x^{k[\nu]}_j$ budeme psát $x^{[\nu]}_j$.

Symbolem $u^{k[\nu]}_{j+\frac{1}{2}}$ (a opět pro jednoduchost vynecháváme v horním indexu k) budeme rozumět aproximaci $u(\ .\ , t_k)|_{[x^{[\nu]}_j, x^{[\nu]}_{j+1}]} \approx u^{[\nu]}_{j+\frac{1}{2}}$, $1 \le j \le n-1$, přesného řešení u rovnice (2.4) na konečném objemu $[x^{[\nu]}_j, x^{[\nu]}_{j+1}]$ příslušnému reálnému času t_k a ν-tému kroku adaptace

sítě (2.18) - (2.19) v čase t_k, tedy $u_{j+\frac{1}{2}}^{[\nu]}$ značí ν-tou iteraci schématu (2.19) s počáteční podmínkou $u_{j+\frac{1}{2}}^{[0]} := u_{j+\frac{1}{2}}^{k}$.

Definujme počáteční rovnoměrné rozložení uzlů x_j^0, $1 \leq j \leq n$, výpočetní sítě ve fyzikální oblasti $\Omega_p = [a, b]$.

Vypočítejme hodnoty $u_{j+\frac{1}{2}}^0$ jako integrální průměr počáteční podmínky $u(x, 0) = u_0(x)$ na příslušném intervalu, tj.

$$u_{j+\frac{1}{2}}^0 = \frac{1}{x_{j+1} - x_j} \int_{x_j}^{x_{j+1}} u_0(x)dx \ , \qquad 1 \leq j \leq n-1.$$

Položme $\nu = 0$ a $k = 0$ a dále $x_j^{[0]} := x_j^0$ pro $1 \leq j \leq n$ a $u_{j+\frac{1}{2}}^{[0]} := u_{j+\frac{1}{2}}^0$ pro $1 \leq j \leq n-1$.

1. **Adaptace sítě**
   Adaptujme síť $\{x_j^{[\nu]}\}$ na novou síť $\{x_j^{[\nu+1]}\}$ podle (2.18) a přepočítejme hodnoty $\{u_{j+\frac{1}{2}}^{[\nu]}\}$ na hodnoty $\{u_{j+\frac{1}{2}}^{[\nu+1]}\}$ pomocí (2.19). Položme $\nu := \nu + 1$ a opakujme tento krok pro předem daný počet iterací m. V praxi volíme počet iterací $m = 1$ - 5. Dostaneme $\{x_j^{[m]}\}$ a $\{u_{j+\frac{1}{2}}^{[m]}\}$.

2. **Výpočet přibližného řešení PDR (2.4)**
   Položme $u_{j+\frac{1}{2}}^{k} := u_{j+\frac{1}{2}}^{[m]}$ pro $1 \leq j \leq n-1$ a $x_j^k := x_j^{[m]}$ pro $1 \leq j \leq n$. Užitím metody konečných objemů vypočítejme na základě hodnot $u_{j+\frac{1}{2}}^{k}$ a sítě $\{x_j^k\}$ přibližné řešení PDR $u_{j+\frac{1}{2}}^{k+1}$ v čase t_{k+1}.

3. **Ukončovací podmínka**
   Jestliže $t_{k+1} \leq T$, definujme $u_{j+\frac{1}{2}}^{[0]} := u_{j+\frac{1}{2}}^{k+1}$ a $x_j^{[0]} := x_j^k$, položme $\nu = 0$, $k = k + 1$ a jděme na krok **1**.

Kapitola 3

Numerické experimenty

V předchozích kapitolách jsme adaptivní metodu zkoumali teoreticky. V této kapitole se zaměříme na praktickou část. Předvedeme si pouze adaptivní metodu (2.18), protože metoda umělého času (2.16) je značně neefektivní a nebudeme se jí tedy zabývat. Pomocí vhodně definovaných funkcí si ukážeme, že se síť adaptuje v závislosti na poloze nespojitosti, tj. že se uzly „shlukují“ v místě nespojitosti, poté adaptivní síť aplikujeme při řešení některých parciálních diferenciálních rovnic a nakonec srovnáme pohyb uzlů adaptivní sítě při použití algoritmu užívajícího schéma (2.18) a algoritmu anizotropního zjemňování výpočetní sítě popsaného v článku [2].

3.1 Funkce s pohybující se nespojitostí

V tomto odstavci budeme vhodnou definicí funkcí, u nichž známe jejich průběh, modelovat situace, kdy se nespojitost funkce v průběhu času pohybuje nebo se velikost skoku v místě nespojitosti zvětšuje či naopak zmenšuje. Na tyto funkce následně aplikujeme metodu adaptivního zjemňování sítě a na obrázcích si ukážeme, jak funguje.

Příklad 3.1. *(Pohybující se nespojitost)*
Definujme si funkci

$$f_1(x,t) = \begin{cases} \begin{cases} x\ , & 0 \le x \le 1+t \\ x-1\ , & 1+t < x \le 3 \end{cases}, & 0 \le t \le 1\ , \\ \begin{cases} x\ , & 0 \le x \le 3-t \\ x-1\ , & 3-t < x \le 3 \end{cases}, & 1 < t \le 3\ . \end{cases}$$

Pro pevné $t \in [0,3]$ je funkce $f_1^t(x) = f_1(x,t)$, $x \in [0,3]$ po částech lineární, přičemž graf funkce f_1^t obsahuje pouze jeden bod nespojitosti, jehož poloha závisí na hodnotě parametru t, parametr t budeme nazývat

čas. Nespojitost v čase $t = 0$ se nachází v bodě $x = 1$ (viz obrázek 3.1), s rostoucím t se posouvá do bodu $x = 2$ pro $t = 1$ a následně se vrací do bodu $x = 1$ v čase $t = 3$.

V analogii ke kapitole 2 funkce $f_1(x, t)$ odpovídá funkci $u(x, t)$ ze zmiňované kapitoly, funkce f_1^t tedy simuluje vývoj přibližného řešení nějaké parciální diferenciální rovnice v reálném čase t, a to následujícím způsobem.

Funkci f_1^t budeme zkoumat v průběhu času pro $t = 0$ až $t = 3$ s časovým krokem 0.1 (což odpovídá časovému kroku při řešení PDR). Máme-li $\mathcal{D}^0$ počáteční rovnoměrnou síť v reálném čase $t = 0$ tvořenou body $x_j,\ 1 \leq j \leq 50$, pak po provedení pěti iterací schématu (2.18) - (2.19) dostaneme novou síť $\mathcal{D}^1$, která vypadá tak, že se uzly sítě posouvají k bodu nespojitosti, jehož poloha odpovídá času $t = 0$ (na obrázku 3.2 ovšem těmto uzlům odpovídá zobrazení uzlů v čase $t = 0.1$, neboť na obrázku znázorňujeme polohu uzlů odpovídající adaptaci sítě na základě hodnot f_1^t z předchozího časového kroku, tedy rozložení uzlů sítě v čase t odpovídá tomu, jakých hodnot funkce f_1^t nabývala v kroku předchozím). Tato síť se následně použije k nalezení přibližného řešení dané PDR, v našem případě jsou ovšem tímto tzv. „řešením“ hodnoty $f_1(x_j, t)$ pro $x_j \in \mathcal{D}^1$ a $t = 0.1$. Nyní opět na síť $\mathcal{D}^1$ aplikujeme pět iterací schématu (2.18) - (2.19) a dostaneme síť $\mathcal{D}^2$, pro kterou zjistíme hodnoty $f_1(x_j, 0.2)$. Další výpočet probíhá analogicky až do $t = 3$.

Výsledný pohyb uzlů je znázorněn na obrázku 3.2. Z něj je patrné, že uzly se shlukují v místě nespojitosti, která se pohybuje v čase, což je přesně to, co od adaptivní metody vyžadujeme.

Příklad 3.2. *(Zvětšující se skok)*
Pro $0 \leq t \leq 1.5$ definujme funkci

$$f_2(x,t) = \begin{cases} (1+t)x & , \quad 0 \ \leq \ x \ \leq \ 0.5 \\ (1+t)x - (1+t) & , \quad 0.5 \ < \ x \ \leq \ 1\ . \end{cases}$$

Pro pevné $t \in [0, 2]$ definujme $f_2^t(x) = f_2(x, t),\ x \in [0, 1]$ Vidíme, že pro $0 \leq t \leq 1.5$ se bod nespojitosti funkce f_2^t nachází neustále v bodě $x = 0.5$, přičemž s rostoucím t se velikost skoku v bodě $x = 0.5$ (tj. pro pevné $t \in [0, 1.5]$ $|\lim_{x \to 0.5_-} f_2^t(x) - \lim_{x \to 0.5_+} f_2^t(x)|$) zvětšuje (viz obrázek 3.3). S tím souvisí i pohyb uzlů sítě na obrázku 3.4, které se se vzrůstajícím časem shlukují u bodu $x = 0.5$. Celkový počet uzlů sítě je 30, časový krok volíme opět 0.1, provádíme 5 iterací adaptivní metody (2.18) - (2.19).

Příklad 3.3. *(Zmenšující se skok)*
Pro $0 \leq t \leq 2$ definujme funkci

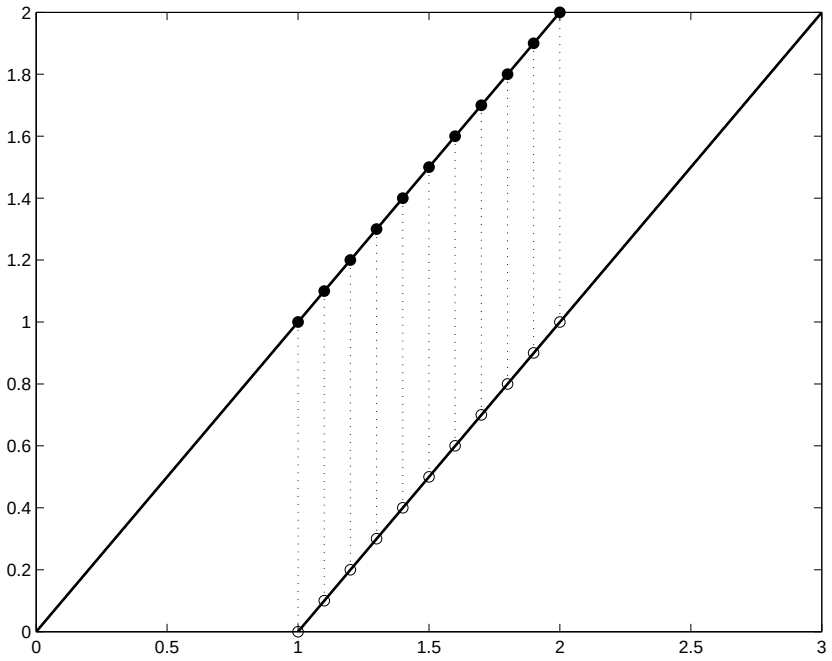
$$f_3(x,t) = \begin{cases} (2-t)x + 0.5t & , \ 0 \leq x \leq 1 \\ (2-t)x - 2 + 1.5t & , \ 1 < x \leq 2 \ . \end{cases}$$

Funkce $f_3^t(x) = f_3(x,t), \ x \in [0,2]$ pro pevné $t \in [0,2]$ se narozdíl od funkce $f_2^t(x)$ chová tak, že se nyní skok mezi hodnotami v bodě $x = 1$ s rostoucím časem nezvětšuje, nýbrž naopak zmenšuje, dokonce v čase $t = 2$ je funkce $f_3^t(x) \equiv 1$ pro $x \in [0,2]$ (viz obrázek 3.5). Uzly se tedy zpočátku pohybují směrem k nespojitosti,v důsledku zmenšování velikosti skoku se však jejich pohyb zpomaluje, dokonce pro $t \to 2$ se uzly pohybují směrem od nespojitosti, velikost skoku už je totiž tak malá, že není nutné v tomto místě síť adaptovat, pohyb uzlů je znázorněn na obrázku 3.6. Počet uzlů sítě je 30, časový krok je opět 0.1, provádíme 4 iterace adaptivní metody (2.18) - (2.19).

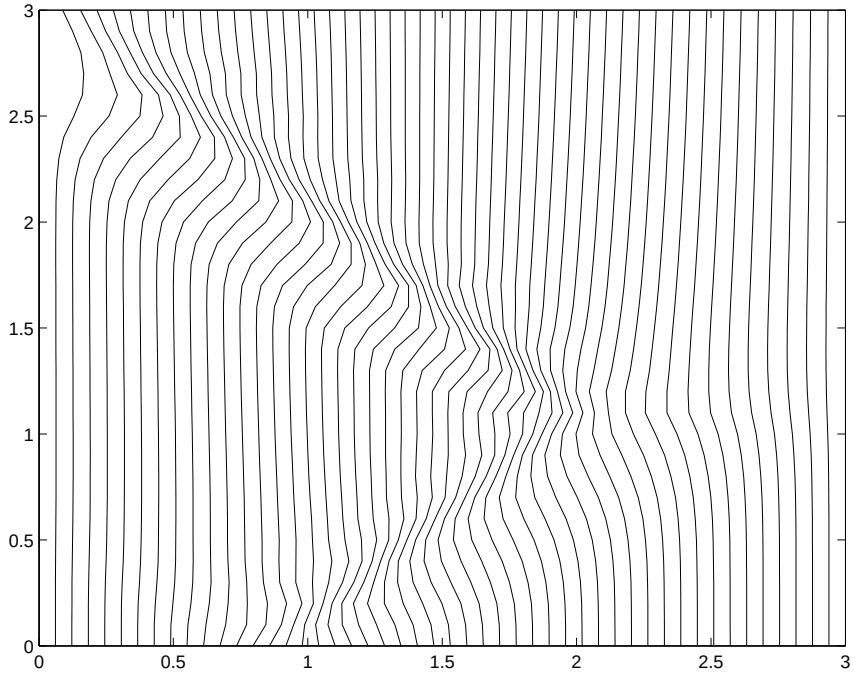
Příklad 3.4. *(Pohybující se nespojitost)*
Pro $0 \leq t \leq 2$ definujme funkci $g(t) = 0.5 + (\exp(-1) - \exp(-1 - t^{0.7}))$ a dále

$$f_4(x,t) = \begin{cases} x & , \ 0 \leq x \leq g(t) \\ x - 1 & , \ g(t) < x \leq 2 \ . \end{cases}$$

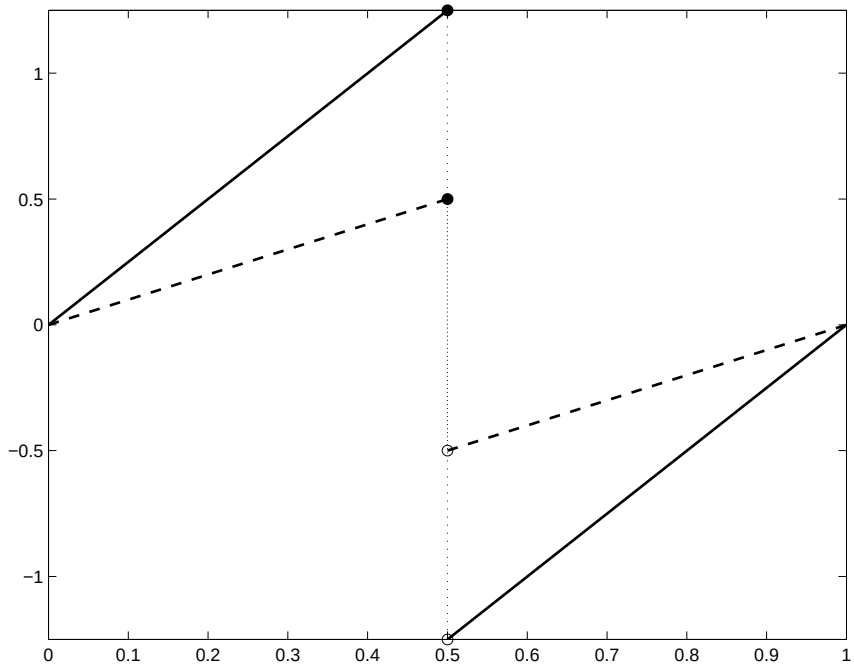
Nespojitost funkce $f_4^t(x) = f_4(x,t), \ x \in [0,2]$, pro pevné $t \in [0,2]$, se nachází v bodě $x = g(t)$ pro $t \in [0,2]$, z čehož je vidět, že se poloha nespojitosti s rostoucím časem mění, avšak rychlost pohybu se pro $t \to 2$ zmenšuje (viz obrázek 3.7), což dobře modeluje situaci, s níž se setkáme například při řešení *Burgersovy rovnice*, a to, že se poloha nespojitosti od určitého časového okamžiku mění už velmi pomalu. Počet uzlů sítě je 50, časový krok je volen 0.1, provádíme 5 iterací adaptivní metody (2.18) - (2.19). Pohyb uzlů sítě viz obrázek 3.8.



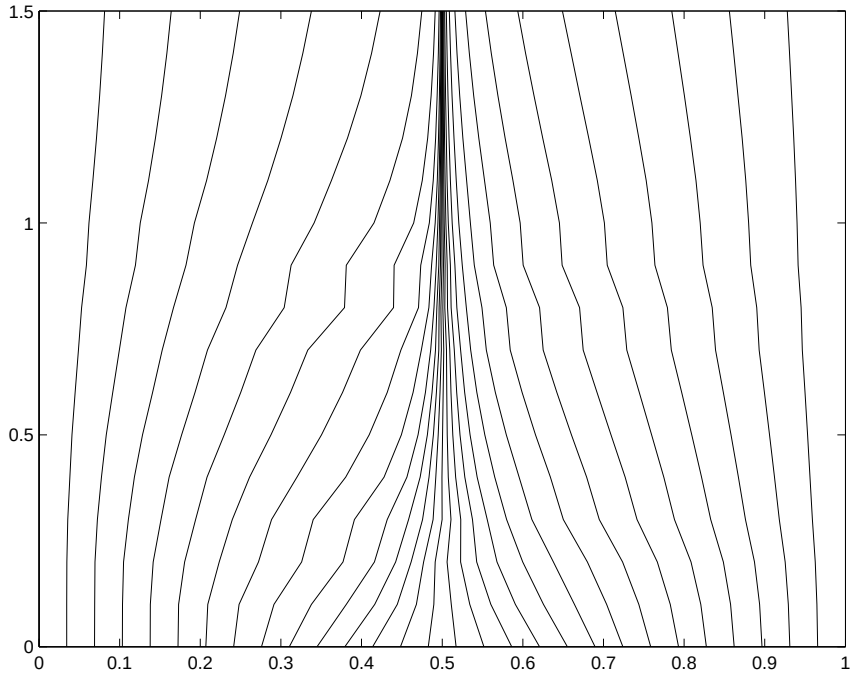
Obrázek 3.1: *Příklad (3.1): Graf funkce* $f_1^{t_k}(x)$ *v čase* $t_k = 0.1k$, $k = 0, 1, 2, \ldots, 10$, *osa* x *znázorňuje hodnoty proměnné* x, *osa* y *hodnoty funkce* $f_1^{t_k}(x)$.



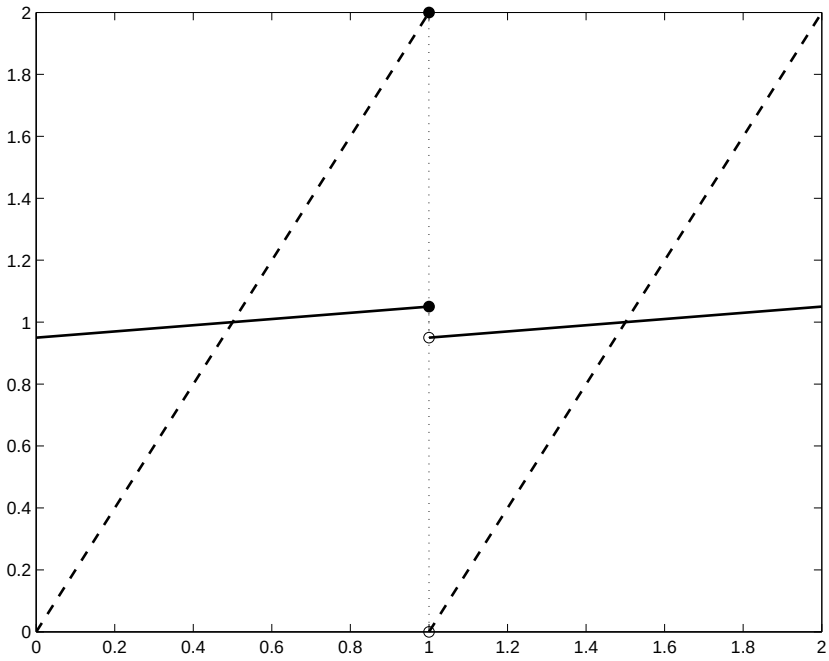
Obrázek 3.2: *Příklad (3.1): Pohyb uzlů sítě v závislosti na pohybující se nespojitosti, osa* x *znázorňuje souřadnice uzlů, osa* y *čas* t.



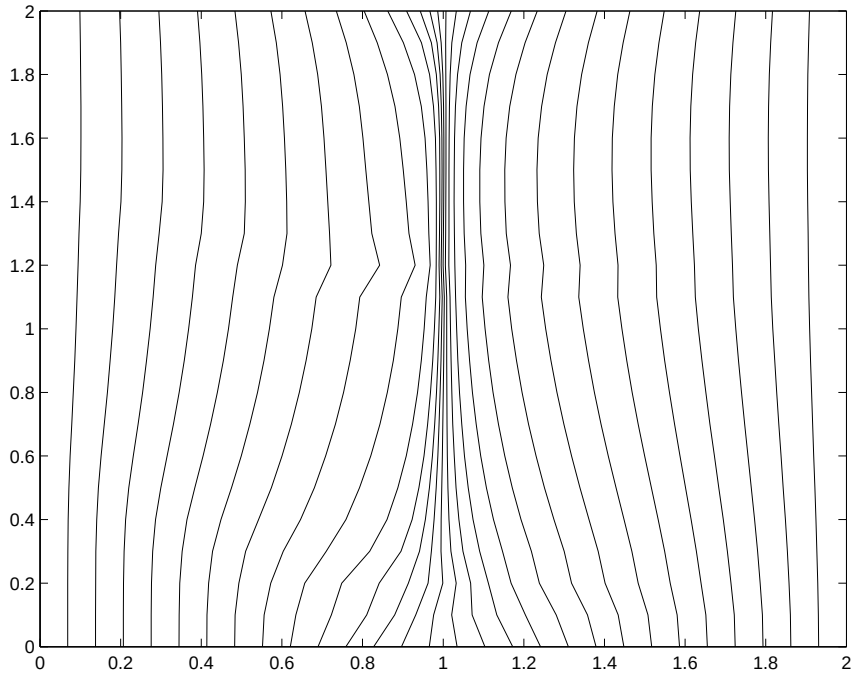
Obrázek 3.3: *Příklad (3.2): Graf funkce* $f_2^t(x)$ *v čase* $t = 0$ *(čárkovaná čára) a v čase* $t = 1.5$ *(plná čára), osa* x *znázorňuje hodnoty proměnné* x*, osa* y *hodnoty funkce* $f_2^t(x)$*.*



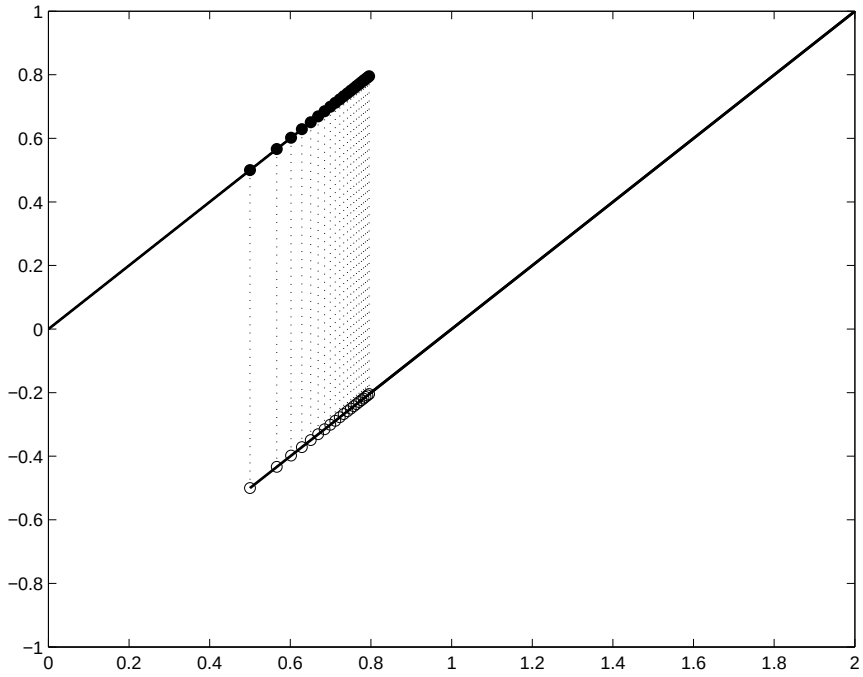
Obrázek 3.4: *Příklad (3.2): Pohyb uzlů sítě v závislosti na zvětšující se velikosti skoku, osa* x *znázorňuje souřadnice uzlů, osa* y *čas* t*.*



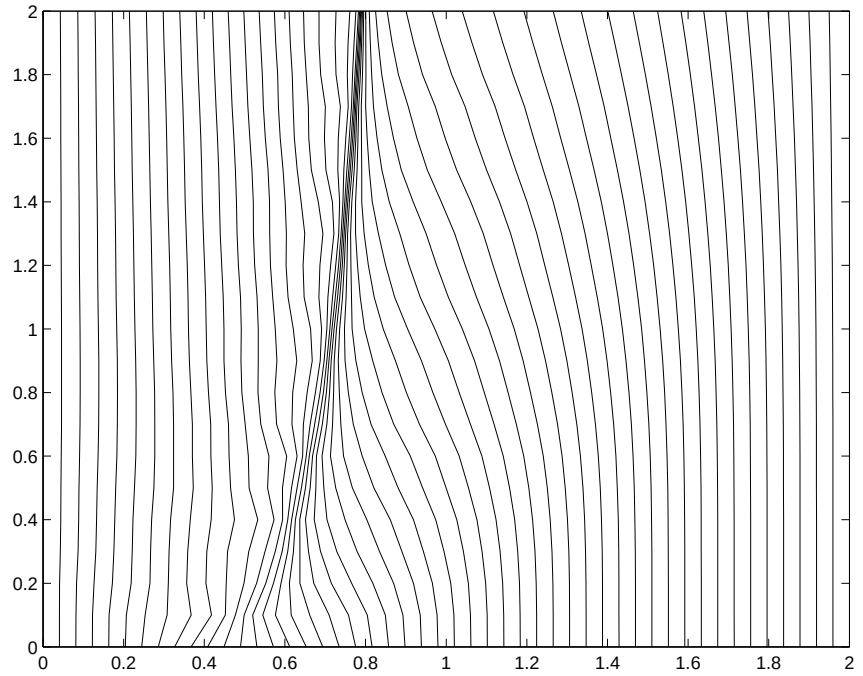
Obrázek 3.5: *Příklad (3.3): Graf funkce* $f_3^t(x)$ *v čase* $t = 0$ *(čárkovaná čára) a v čase* $t = 1.9$ *(plná čára), osa* x *znázorňuje hodnoty proměnné* x*, osa* y *hodnoty funkce* $f_3^t(x)$*.*



Obrázek 3.6: *Příklad (3.3): Pohyb uzlů sítě v závislosti na zmenšující se velikosti skoku, osa* x *znázorňuje souřadnice uzlů, osa* y *čas* t*.*



Obrázek 3.7: *Příklad (3.4): Graf funkce* $f_4^{t_k}(x)$ *v čase* $t_k = 0.1k$, $k = 0, 1, 2, \ldots, 20$, *osa* x *znázorňuje hodnoty proměnné* x, *osa* y *hodnoty funkce* $f_4^{t_k}(x)$.



Obrázek 3.8: *Příklad (3.4): Pohyb uzlů sítě v závislosti na pohybující se nespojitosti, osa* x *znázorňuje souřadnice uzlů, osa* y *čas* t.

3.2 Aplikace adaptivní sítě při řešení PDR

V této části se zaměříme na aplikaci adaptivní metody při řešení parciálních diferenciálních rovnic a ukážeme si na konkrétních příkladech, jak adaptivita výpočetní sítě zvyšuje přesnost přibližného řešení ve srovnání s použitím rovnoměrné sítě během celého výpočtu. Budeme hledat přibližné řešení úlohy (2.4) s počáteční podmínkou (2.5).

Pro nalezení přibližného řešení rovnice (2.4) použijeme tzv. MUSCL (monotone upstream-centered scheme for conservation laws) metodu konečných objemů, jejíž definici a schéma algoritmu lze nalézt na str. 494 článku [3].

Podívejme se již na konkrétní příklady.

3.2.1 Burgersova rovnice

Předpis *Burgersovy rovnice* je následující:

$$\frac{\partial u}{\partial t}(x,t) + \frac{\partial\,(\frac{u^2}{2})}{\partial x}(x,t) = 0, \quad t > 0, \quad 0 \le x \le 2\pi, \tag{3.1}$$

se spojitou počáteční podmínkou

$$u(x,0) = 0.5 + sin(x), \quad x \in [0, 2\pi], \tag{3.2}$$

a periodickou okrajovou podmínkou.

Užitím MUSCL algoritmu s adaptivním zjemňováním sítě metodou (2.18) - (2.19) splňující geometrický zákon zachování jsme získali přibližné řešení rovnice (3.1) v čase $t = 2$, které je na obrázku 3.9 znázorněno pomocí bodů, přesné řešení je znázorněno plnou čarou. Graf přesného řešení byl získán *metodou charakteristik*. Na obrázku 3.10 je zobrazeno přibližné řešení, které jsme získali stejným MUSCL algoritmem za užití stejných parametrů, avšak bez zjemňování sítě, tzn. že během celého výpočtu jsme pracovali na rovnoměrné síti na intervalu $[0, 2\pi]$. V místech, na jejichž okolí je přesné řešení hladké, dostáváme při použití obou postupů obdobné hodnoty přibližného řešení, avšak v místě nespojitosti je přibližné řešení získané pomocí zjemňování sítě mnohem přesnější (ve smyslu, kdy přesnost určujeme jako absolutní hodnotu rozdílu funkčních hodnot přesného a přibližného řešení), narozdíl od rovnoměrné sítě, kdy na okolí bodu nespojitosti dochází k výraznému zkreslení přibližného řešení.

Pohyb uzlů adaptivní sítě je zachycen na obrázku 3.11. V průběhu výpočtu dochází k jejich postupnému shromažďování kolem bodu nespojitosti a přibližně od času $t = 1.3$ už jen nejbližší skupina uzlů sleduje pohyb nespojitosti, uzly, které jsou dostatečně daleko od bodu nespojitosti, se téměř nepohybují a zachovávají takřka rovnoměrné rozložení, neboť

gradient přibližného řešení je vzhledem k bodu nespojitosti v ostatních bodech značně malý.

Výpočetní síť byla v tomto případě tvořena 30-ti uzly, v každém časovém kroku MUSCL algoritmu jsme použili 5 iterací schématu (2.18) - (2.19) pro zjemnění sítě, časový krok MUSCL metody byl volen 0.0005.

3.2.2 Riemannův problém

Rovnice pro tzv. Riemannův problém pro hyperbolickou rovnici zákona zachování je tvaru

$$\frac{\partial u}{\partial t}(x,t) + \frac{\partial\,(f(u))}{\partial x}(x,t) = 0, \quad t > 0, \quad -1 \le x \le 1, \tag{3.3}$$

kde

$$f(u) = \frac{1}{4}(u^2-1)(u^2-4),$$

s nespojitou počáteční podmínkou

$$u(x,0) = -2\,\mathrm{sgn}(x), \quad -1 \le x \le 1. \tag{3.4}$$

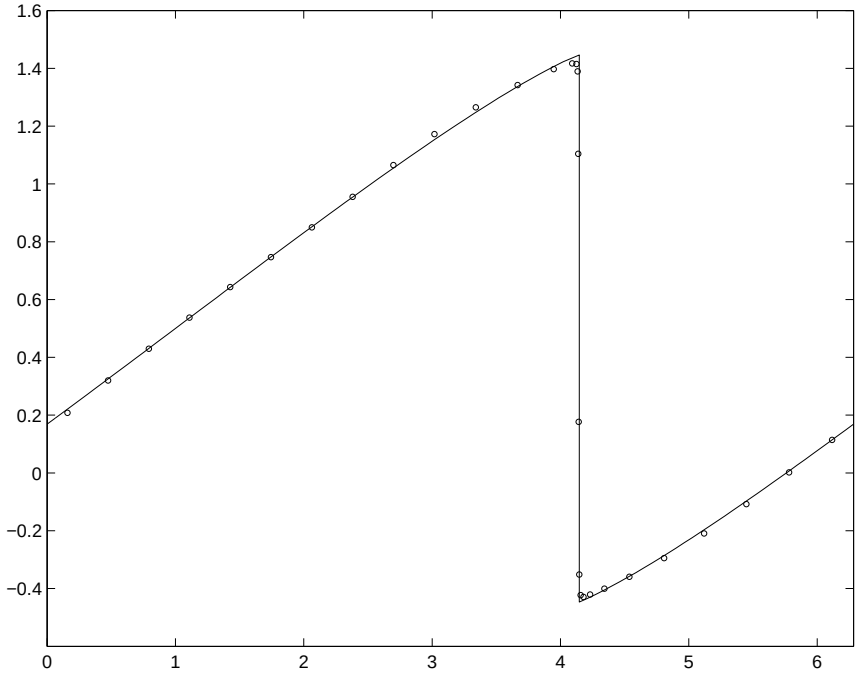
a okrajovou podmínkou

$$\begin{aligned} u(-1,t) &= -2, \quad t > 0, \\ u(1,t) &= \quad 2, \quad t > 0. \end{aligned} \tag{3.5}$$

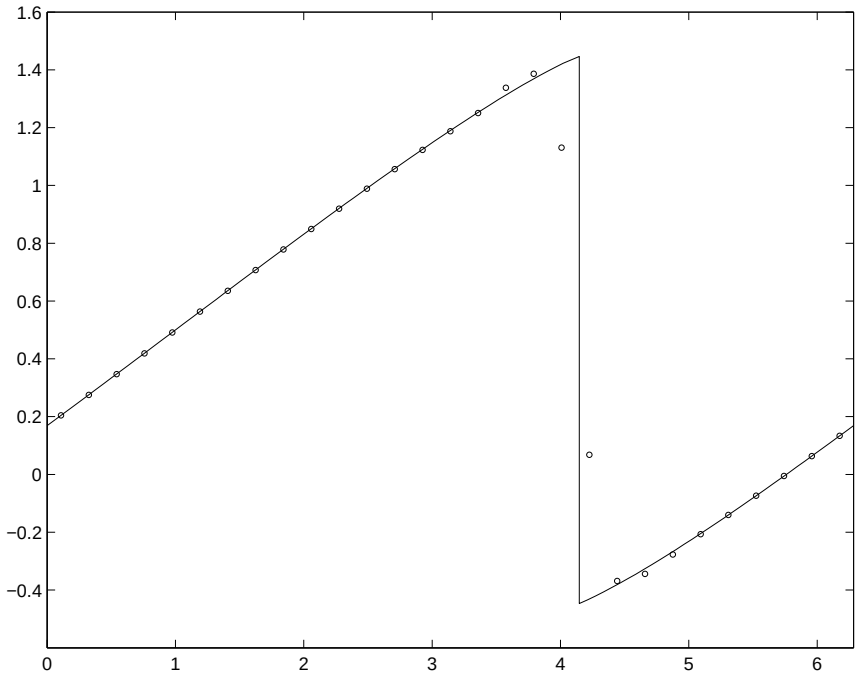
Tato parciální diferenciální rovnice a graf jejího přesného řešení jsou převzaty z článku [3]. Přesné řešení v případě této rovnice není spojité již od času $t = 0$, jak je vidět z definice počáteční podmínky a v čase $t = 1.2$ obsahuje dva body nespojitosti.

Opět jsme použili MUSCL algoritmus k nalezení přibližného řešení rovnice (3.3). Na výpočetní síti, která je tvořena 50-ti uzly, jsme opět pomocí adaptivního zjemňování získali mnohem přesnější řešení než v případě použití pouze rovnoměrné sítě po celou dobu výpočtu, jak je vidět z obrázků 3.13 a 3.14. Ke zkreslení přibližného řešení v případě užití rovnoměrné sítě dochází opět pouze v místech nespojitosti přesného řešení. V případě adaptivní sítě se uzly akumulují na okolí místa nespojitosti a absolutní hodnota rozdílu hodnot přibližného a přesného řešení je menší než v případě užití pouze rovnoměrné sítě. Pohyb uzlů výpočetní sítě je znázorněn na obrázku 3.12.

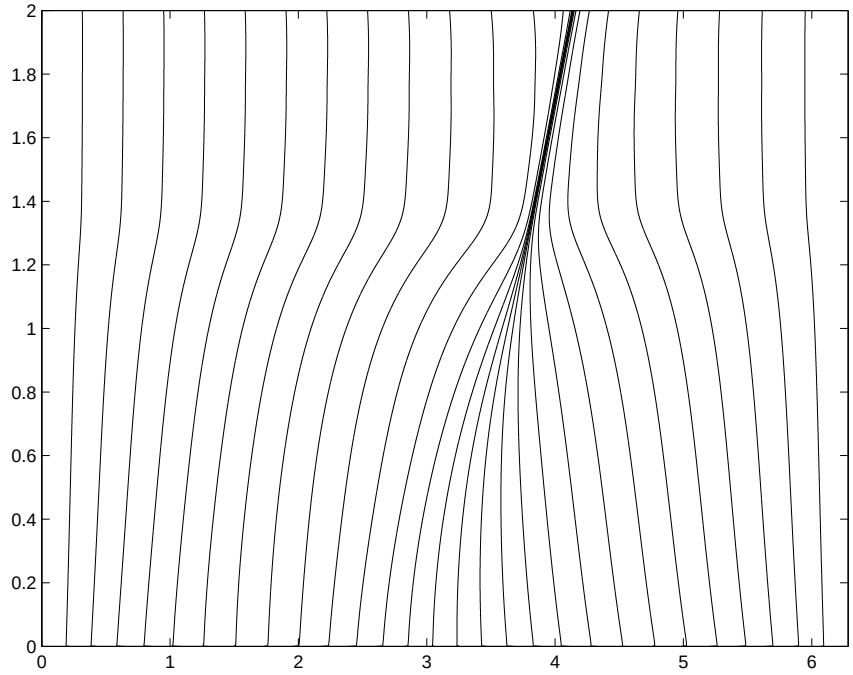
V každém časovém kroku MUSCL algoritmu jsme adaptovali výpočetní síť pouze pomocí jedné iterace schématu (2.18) - (2.19), avšak použili jsme modifikovanou monitorovací funkci $\omega^\theta = \sqrt{1+\theta|u_x|^2}$ s parametrem $\theta > 0$, v našem případě jsme volili $\theta = 0.6$. Časový krok MUSCL algoritmu byl volen 0.0001.



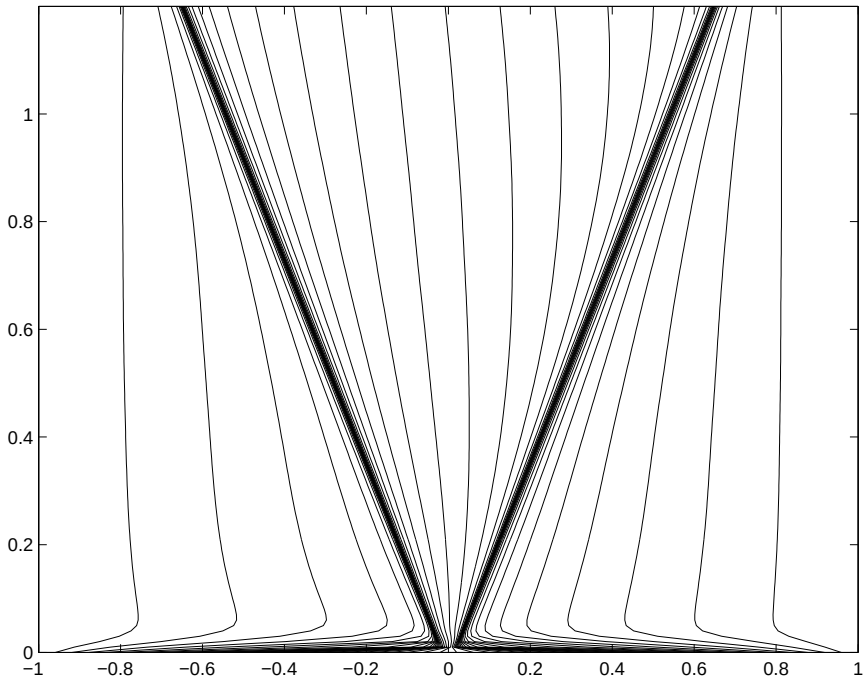
Obrázek 3.9: *Graf přesného (plná čára) a přibližného (body) řešení Burgersovy rovnice (3.1) na intervalu* $[0, 2\pi]$ *v čase* $t = 2$ *získaného pomocí adaptivního zjemňování výpočetní sítě.*



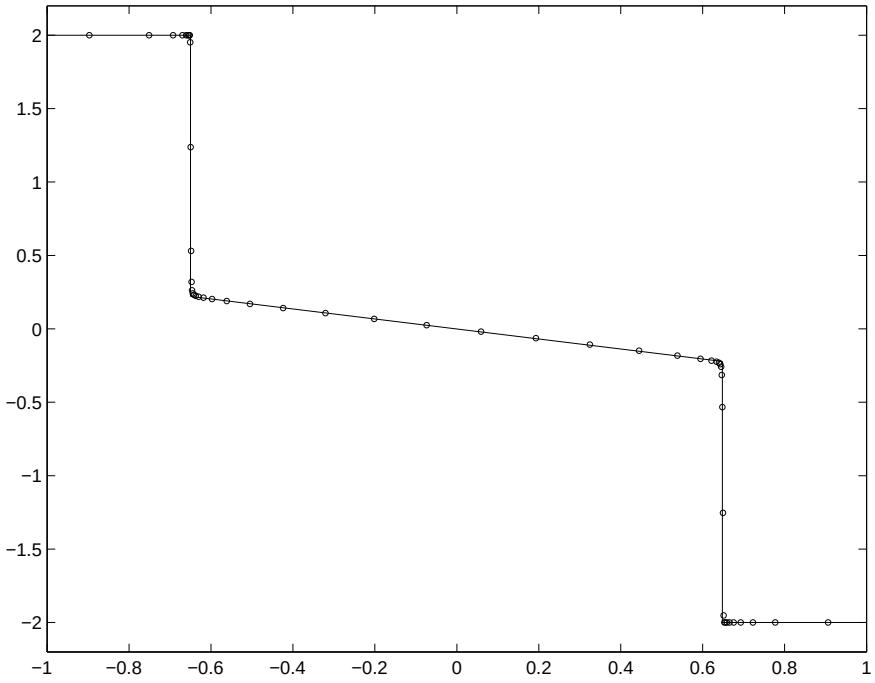
Obrázek 3.10: *Graf přesného (plná čára) a přibližného (body) řešení Burgersovy rovnice (3.1) na intervalu* $[0, 2\pi]$ *v čase* $t = 2$ *získaného na rovnoměrné výpočetní síti.*



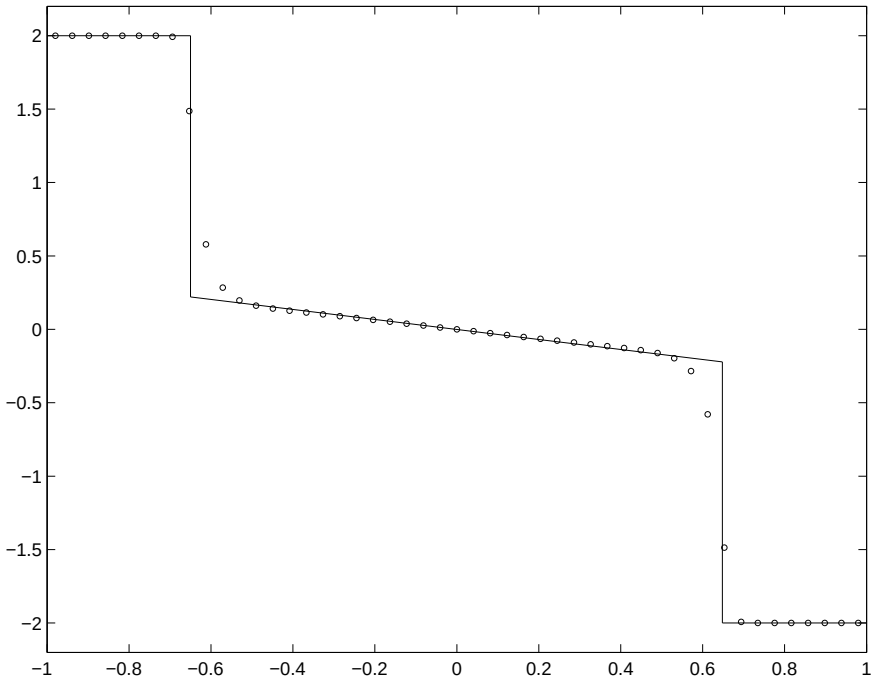
Obrázek 3.11: *Pohyb uzlů výpočetní sítě v závislosti na čase t během hledání přibližného řešení Burgersovy rovnice (3.1) na intervalu* $[0, 2\pi]$. *Osa x reprezentuje souřadnice uzlů, osa y čas t.*



Obrázek 3.12: *Pohyb uzlů výpočetní sítě v závislosti na čase t během hledání přibližného řešení rovnice (3.3) na intervalu* $[-1, 1]$. *Osa x reprezentuje souřadnice uzlů, osa y čas t.*



Obrázek 3.13: *Graf přesného (plná čára) a přibližného (body) řešení rovnice (3.3) na intervalu* $[-1, 1]$ *v čase* $t = 1.2$ *získaného pomocí adaptivního zjemňování výpočetní sítě.*



Obrázek 3.14: *Graf přesného (plná čára) a přibližného (body) řešení rovnice (3.3) na intervalu* $[-1, 1]$ *v čase* $t = 1.2$ *získaného na rovnoměrné výpočetní síti.*

3.3 Srovnání různých adaptivních metod

Na závěr srovnáme adaptivní metodu zkoumanou v této práci a anizotropní adaptivní metodu z článku [2] stejně jako v odstavci 3.1 pro funkce s nespojitostí pohybující se v čase.

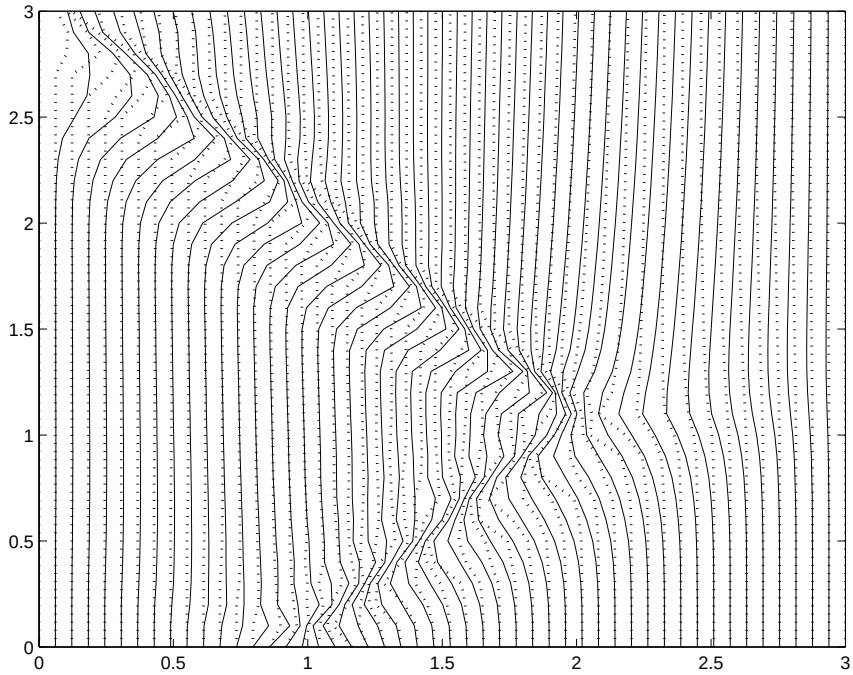
Z obrázků 3.15 - 3.18 je patrné, že u obou metod dochází ke zhuštění sítě v místě nespojitosti a toto „zhuštění“ prvků sítě se pohybuje spolu s nespojitostí. Na zbytku výpočetní sítě obě adaptivní metody zachovávají rovnoměrné dělení, nedochází k pohybu uzlů v místech, kde není potřeba síť zjemnit. V tomto ohledu tedy obě metody vykazují obdobné vlastnosti a jsou vhodné k použití například při řešení parciálních diferenciálních rovnic, jak jsme viděli na předcházejících příkladech. Ze srovnání výpočetní náročnosti vychází jako rychlejší, zhruba o jeden řád, anizotropní adaptivní metoda, viz tabulka 3.1. Časovou náročnost jsme zkoumali v následujícím smyslu.

Pro každý algoritmus zvlášť jsme během příslušného časového intervalu adaptovali výpočetní síť pro funkce definované v příkladech (3.1) - (3.4), v případě adaptivní metody (2.18) - (2.19) jsme volili stejné parametry jako v příkladech (3.1) - (3.4), v případě anizotropní adaptivní metody jsme na stejném výpočetním intervalu během stejného časového intervalu volili stejný počet uzlů jako u předcházející metody, parametry byly voleny takto: MAXITER = 1, e1 = 100, e2 = 1 (význam parametrů viz přiložené CD a internetové stránky http://www.karlin.mff.cuni.cz/~dolejsi/angen/angen3.1.htm). Tento krok výpočtu sítě jsme opakovali desetkrát a výsledný čas, během kterého proběhl výpočet, jsme vydělili deseti a tak získali hodnoty uvedené v tabulce 3.1. Pro určení výpočetního času jsme užili standardní matlabovské příkazy `tic` a `toc`, výsledné časy jsou uvedené v sekundách.

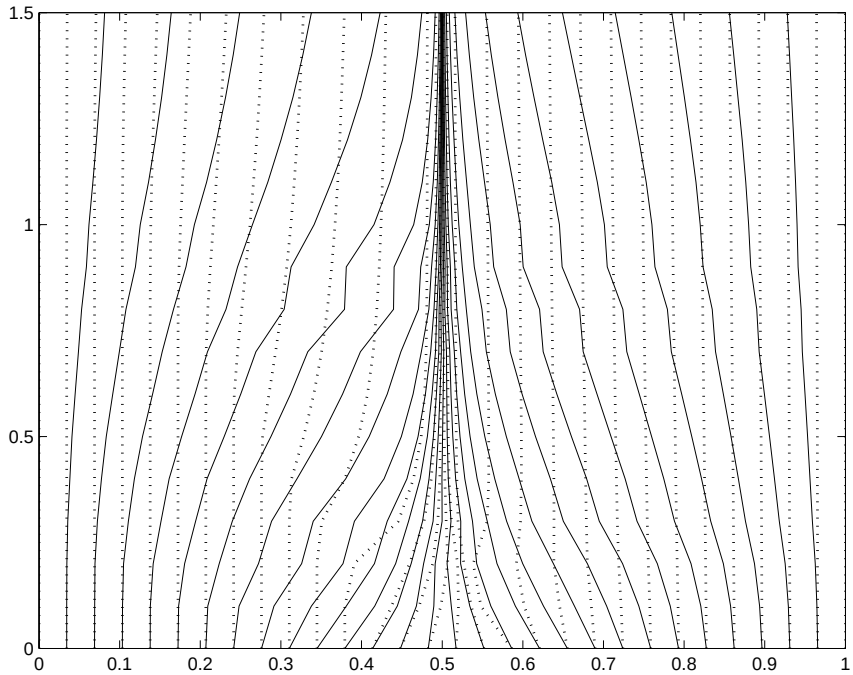
Zkoumání výpočetní náročnosti jsme prováděli na počítači Intel(R) Pentium(R) M, 1.73GHz / 504 MB RAM v programu Matlab verze 7.2.0.232 (R2006a).

Funkce	Schéma (2.18) - (2.19)	Anizotropie
f_1	0.2238	0.0227
f_2	0.0791	0.0077
f_3	0.0938	0.0097
f_4	0.1080	0.0105

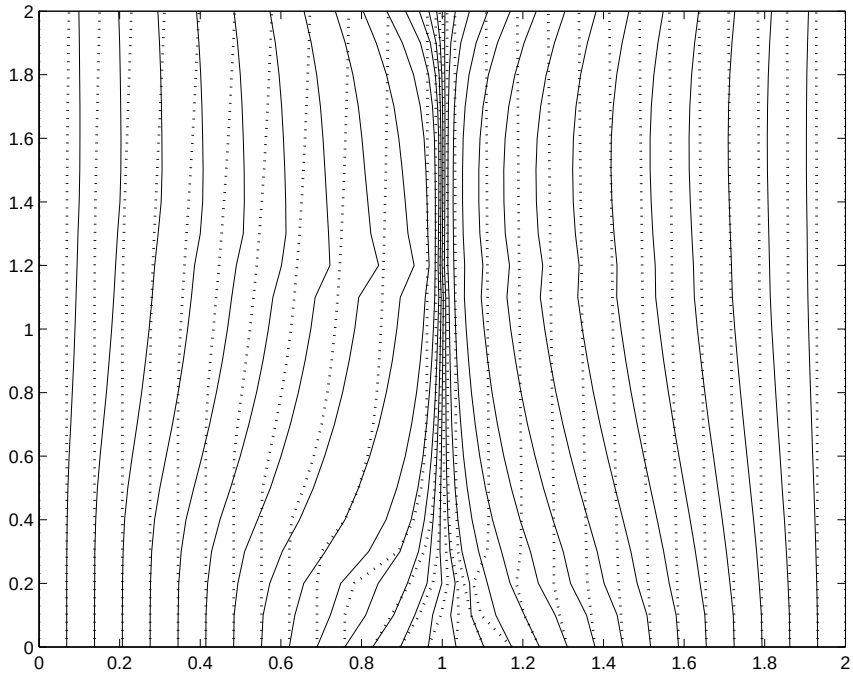
Tabulka 3.1: Srovnání časové náročnosti jednotlivých algoritmů při aplikaci pro funkce definované v příkladech (3.1) - (3.4). V levém sloupci je uvedena daná funkce, v prostředním sloupci čas schématu (2.18) - (2.19) a v pravém sloupci čas anizotropní adaptivní metody, časy jsou uvedeny v sekundách.



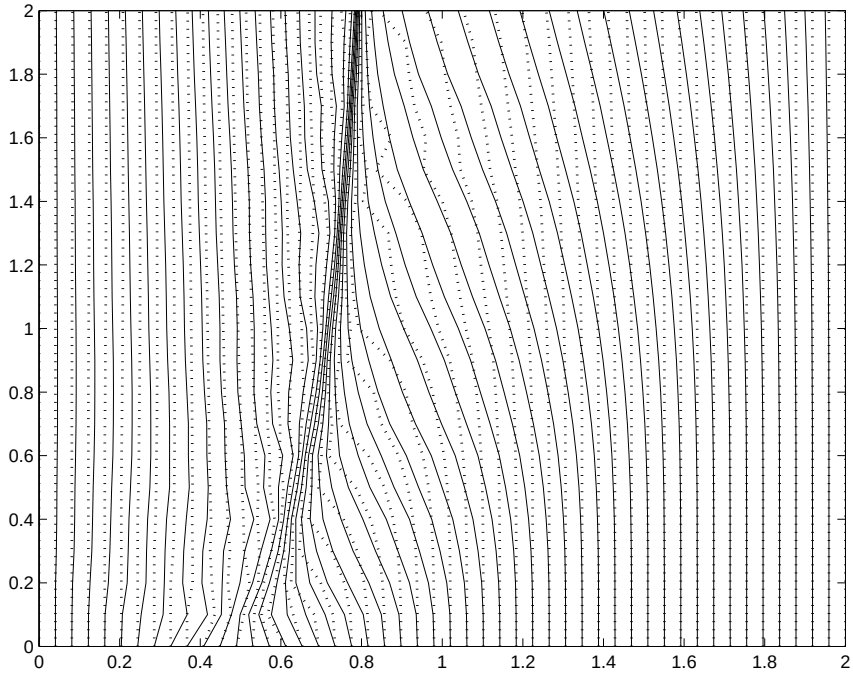
Obrázek 3.15: *Funkce z př. (3.1): Plná čára znázorňuje pohyb uzlů metody (2.18), tečkovaná pohyb uzlů anizotropní adaptivní metody, osa x reprezentuje souřadnice uzlů, osa y čas t.*



Obrázek 3.16: *Funkce z př. (3.2): Plná čára znázorňuje pohyb uzlů metody (2.18), tečkovaná pohyb uzlů anizotropní adaptivní metody, osa x reprezentuje souřadnice uzlů, osa y čas t.*



Obrázek 3.17: *Funkce z př. (3.3): Plná čára znázorňuje pohyb uzlů metody (2.18), tečkovaná pohyb uzlů anizotropní adaptivní metody, osa x reprezentuje souřadnice uzlů, osa y čas t.*



Obrázek 3.18: *Funkce z př. (3.4): Plná čára znázorňuje pohyb uzlů metody (2.18), tečkovaná pohyb uzlů anizotropní adaptivní metody, osa x reprezentuje souřadnice uzlů, osa y čas t.*

Literatura

[1] M. Feistauer, J. Felcman, and I. Straškraba. *Mathematical and Computational Methods for Compressible Flow*. Oxford University Press, Oxford, 2003.

[2] J. Felcman and P. Kubera. Computational aspects of the mesh adaptation for the time marching procedure. In A. Bermúdez de Castro, D. Gómez, P. Quintela, and P. Salgado, editors, *Numerical Mathematics and Advanced Applications, ENUMATH 2005*, pages 225–232, Santiago de Compostela, Spain, July 2005 2006. Springer Verlag Berlin Heidelberg. ISBN-10 3-54034287-7 Springer Berlin Heidelberg New York, ISBN-13 978-3-540-34287-8 Springer Berlin Heidelberg New York.

[3] H. Tang and T. Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J. Appl. Math.*, pages 487–515, 2003.