

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2009

Vojtěch Sokol

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

**Metoda konečných prvků pro rovnice
povrchového proudění**

**A Finite Element Method for Shallow
Water Equations**

2009

Vojtěch Sokol

vložit zadání

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. května 2009

.....
Vojtěch Sokol

Chtěl bych poděkovat vedoucímu diplomové práce Ing. Dalibor Lukášovi, Ph.D. za trpělivost a poskytování cenných rad které mě nakonec dovedly na správnou cestu. Dále pak rodině za podporu, přítelkyni za porozumění a všem kteří mě během psaní této práce drželi při smyslech. V neposlední řadě bych chtěl poděkovat celé Katedře aplikované matematiky a zvláště pak Doc. RNDr. Jiřímu Bouchalovi, Ph.D. za vynikající výuku matematiky.

Abstrakt

Tato práce se zabývá numerickým řešením Saint-Venantových rovnic, které popisují proudění vody v říčních tocích, metodou konečných prvků (MKP). Rovnice odvodíme z druhého Newtonova zákona a zákona zachování hmoty. MKP diskretizace bude provedena pro lineárně-konstantní a kvadraticko-lineární prvky. Diskretizací dostaneme soustavu nelineárních algebraických rovnic, které dále řešíme Newtonovou metodou spolu s iterační metodou GMRES.

Klíčová slova

Saint-Venantovy rovnice, rovnice mělké vody, metoda konečných prvků, Newtonova metoda, GMRES

Abstract

This thesis deals with numerical solution of Saint-Venant equations, which describes water flow in rivers, by finite element method (FEM). Equations will be derived from second Newton law and law of mass conservation. FEM discretisation will be done using quadratic-linear and linear-constant elements. After discretisation we get system of nonlinear algebraic equations. For their solution we use Newton method and iterative GMRES method.

Keywords

Saint-Venant equations, shallow water equations, finite element method, Newton method, GMRES

Seznam použitých symbolů a zkratek

a	zrychlení [m s^{-2}]
g	tíhové zrychlení [m s^{-2}]
h	výška vodního sloupce [m]
l	délka koryta [m]
m	hmotnost [kg]
p	tlak [Pa]
t	čas [s]
u	rychlost [m s^{-1}]
x	prostorová proměnná [m]
Δx	délka prostorového kroku [m]
Δt	délka časového kroku [s]
$\delta_{i,j}$	Kroneckerovo delta
ν	kinematická viskozita [$\text{m}^2 \text{s}^{-1}$]
ρ	hustota [kg m^{-3}]

Obsah

1	Úvod.....	3
2	Fyzikální model.....	4
2.1	Saint-Venantovy rovnice.....	4
2.1.1	Pohybová rovnice.....	4
2.1.2	Rovnice kontinuity.....	6
2.2	Počátečně-okrajová úloha.....	7
3	Slabá formulace, diskretizace.....	9
3.1	Časová diskretizace.....	9
3.2	Slabá formulace.....	9
3.3	Prostorová diskretizace a MKP.....	11
3.4	Vyčíslení MKP matic.....	15
4	Numerické řešení.....	20
4.1	Newtonova metoda.....	20
4.2	GMRES.....	23
5	Numerické simulace.....	26
5.1	Simulace s lineárně-konstantními prvky.....	26
5.2	Simulace s kvadraticko-lineárními prvky.....	29
6	Závěr.....	32
7	Literatura.....	33
	Přílohy.....	34

Seznam obrázků

Obrázek 2.1: Síly působící na kontrolní element.....	6
Obrázek 2.2: Interpretace proměnných v 1D.....	7
Obrázek 2.3: Počátečně-okrajová podmínka systému rovnic.....	8
Obrázek 3.1: Po částech lineární básová funkce.....	12
Obrázek 3.2: Po částech kvadratické básové funkce.....	13
Obrázek 3.3: Zobrazení F zobrazující referenční element na element e_i	17
Obrázek 4.1: Ilustrace k nutnosti bisekce v Newtonově metodě.....	22

1 Úvod

Cílem této práce je numerické řešení rovnic povrchového proudění metodou konečných prvků. Tyto rovnice popisují proudění vody v říčních tocích, jejich efektivní řešení je tedy významné například pro simulaci šíření povodňových vln.

Práce bude rozdělena do následujících kapitol:

Ve druhé kapitole se budeme zabývat odvozením Saint-Venantových rovnic (rovnice mělké vody), které popisují proudění kapaliny v říčním korytě. Rovnice budou odvozeny z 2. Newtonova zákona a ze zákona zachování hmoty.

Třetí kapitola je věnována slabé formulaci problému, jeho časové diskretizaci pomocí dopředné Eulerovy metody a prostorové diskretizaci metodou konečných prvků (MKP). V této kapitole je dále popis sestavení MPK matic po elementech lokálních matic a také odvození vyčíslení lokálních matic zobrazením z referenčního elementu.

Čtvrtá kapitola se zabývá numerickým řešením vzniklých nelineárních rovnic pomocí Newtonovy metody s využitím iterační metody GMRES pro řešení lineárních rovnic v rámci Newtonovy metody.

Nakonec v páté kapitole se budeme věnovat výsledkům numerických simulací pro lineárně-konstantní a kvadraticky-lineární prvky.

2 Fyzikální model

2.1 Saint-Venantovy rovnice

Saint-Venantovy rovnice (neboli rovnice mělké vody) se používají k popisu proudění vody říčním korytem. Použití těchto rovnic je omezeno těmito předpoklady:

- 1) hustota kapaliny je konstantní,
- 2) poměr hloubky vody k délce popisovaného úseku řeky je malý, můžeme tedy zanedbat vertikální složky rychlosti u a její velikost uvažujeme konstantní vzhledem k hloubce h ,
- 3) vertikální komponenta zrychlení je zanedbatelná (tlak je možno uvažovat hydrostatický),
- 4) v našem modelu neuvažujeme působení dalších vlivů, jako je například Coriolisova síla, tření o dno, vítr, změny množství kapaliny způsobené srážkami případně odpařováním.

Saint-Venantovy rovnice lze odvodit zjednodušením z Navier-Stokesových rovnic (odvození viz [5]), jednodušší je však využít k odvození 2. Newtonův zákon pro pohybovou rovnici a zákon zachování hmoty (nebo Reynoldsův transportní teorém) pro rovnici kontinuity.

2.1.1 Pohybová rovnice

K odvození pohybové rovnice využijeme 2. Newtonův zákon - zákon síly:

$$F = m a = m \frac{du}{dt}, \quad (2.1)$$

kde F síla působící na element kapaliny výraz $m a$ je setrvačná síla elementu. Časovou derivaci rychlosti rozepíšeme jako derivaci složené funkce, neboť závisí jak na prostorové proměnné x , tak na časové proměnné t :

$$\frac{du}{dt} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} u + \frac{\partial u}{\partial t}, \quad (2.2)$$

a dosadíme do (2.1):

$$F = m \left(\frac{\partial u}{\partial x} u + \frac{\partial u}{\partial t} \right), \quad (2.3)$$

Neuvažujeme-li žádné dodatečné síly (například tření kapaliny o hranici), má působící síla F dvě složky: tlakovou a gravitační část. Uvažujeme-li dále podloží kolmé na směr působení gravitační síly, nemá gravitační síla vliv na pohyb tekutiny, a zbývá tedy pouze tlaková část způsobená tlakem na hranici elementu kapaliny a síla způsobená změnou průřezu koryta A v rámci elementu kapaliny:

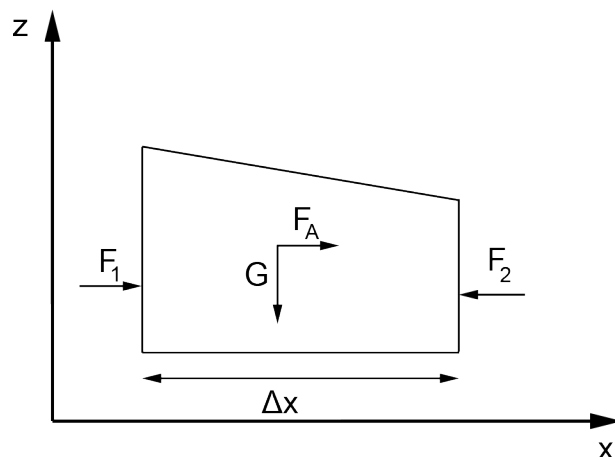
$$\begin{aligned} F &= F_1 - F_2 + F_A \\ F_1 &= pA, F_2 = pA + \frac{\partial(pA)}{\partial x} \Delta x, \\ F_A &= p \frac{\partial A}{\partial x} \Delta x \\ F &= -\frac{\partial p}{\partial x} A \Delta x. \end{aligned} \quad (2.4)$$

Dále si v rovnici (2.3) vyjádříme hmotnost m pomocí průřezu A a dosadíme z rovnice (2.4):

$$\rho A \Delta x \left(\frac{\partial u}{\partial x} u + \frac{\partial u}{\partial t} \right) = -\frac{\partial p}{\partial x} A \Delta x. \quad (2.5)$$

Za předpokladu hydrostatického tlaku $p = h\rho g$ spolu s nestlačitelností kapaliny ($\rho = \text{konst.}$) dostaneme po úpravě pohybovou rovnici

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} u + g \frac{\partial h}{\partial x} = 0. \quad (2.6)$$



Obrázek 2.1: Síly působící na kontrolní element

K rovnici (2.6) ještě doplníme viskózní člen, jehož odvození z Navier-Stokesových rovnic lze nalézt v [5]. Výsledná rovnice tedy bude vypadat takto:

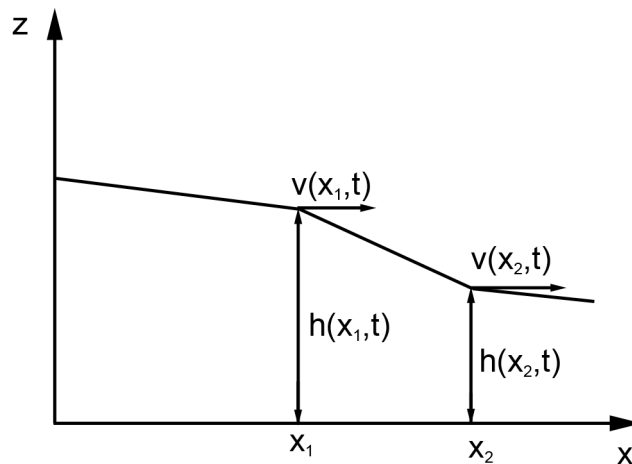
$$\frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} u + g \frac{\partial h}{\partial x} = 0. \quad (2.7)$$

2.1.2 Rovnice kontinuity

K odvození rovnice kontinuity využijeme zákona zachování hmoty

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho h dx + \int_{x_1}^{x_2} \frac{\partial}{\partial x} (\rho h u) dx = 0, \quad (2.8)$$

který říká, že součet změny hmotnosti elementu kapaliny (tedy oblast ohraničená prostorovými proměnnými x_1, x_2) a změny hmotnosti způsobené vtečením resp. vytečením hranic elementu je nulový. Situaci ilustruje obrázek 2.2.



Obrázek 2.2: Interpretace proměnných v 1D

Za předpokladu nestlačitelnosti kapaliny můžeme z rovnice (2.8) opět vyloučit hustotu ρ . Jsou-li funkce vystupující jako integrandy rovnice (2.8) dostatečně hladké, lze zaměnit pořadí integrace a derivace¹

$$\int_{x_1}^{x_2} \left(\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} \right) dx = 0. \quad (2.9)$$

Jelikož integrační meze lze v kapalině volit libovolně, musí být integrand roven identicky nule, čímž dostaneme rovnici kontinuity:

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} = 0. \quad (2.10)$$

2.2 Počátečně-okrajová úloha

V dalším textu budeme uvažovat následující systém rovnic

¹ Necht' $\left\{ f(x, y), \frac{\partial f}{\partial y} \right\} \in C(\langle a, b \rangle \times \langle c, d \rangle)$, $g(y) = \int_a^b f(x, y) dx$.

Pak existuje $\frac{dg}{dy}$, a platí: $\frac{dg}{dy} = \frac{d}{dy} \int_a^b f(x, y) dx = \int_a^b \frac{\partial f}{\partial y}(x, y) dx$ viz [2, str. 510].

$$\frac{\partial u}{\partial t} - v \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} u + g \frac{\partial h}{\partial x} = 0, \quad (2.11)$$

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} = 0,$$

uzavřený okrajovými

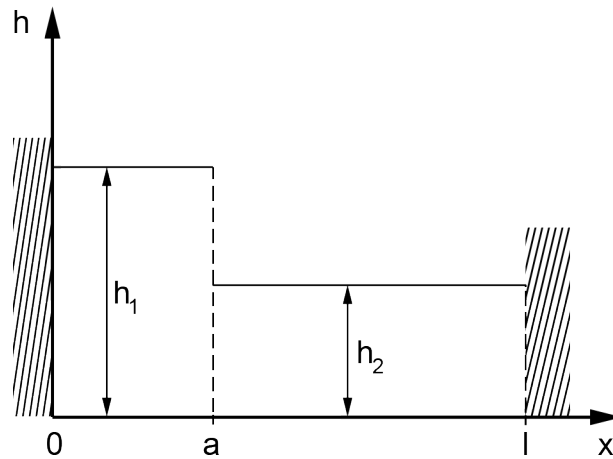
$$u(x,0) = 0 \quad \forall x \in [0, l],$$

$$h(x,0) = \begin{cases} h_1; & x < a \\ h_2; & x \geq a \end{cases} \quad (2.12)$$

a počátečními podmínkami

$$u(0,t) = u(l,t) = 0 \quad \forall t \in [0, T]. \quad (2.13)$$

Tento systém popisuje zjednodušený jedno-dimenzionální model šíření vlny v korytě řeky délky l při protržení přehrady, viz obrázek 2.3.



Obrázek 2.3: Počátečně-okrajová podmínka systému rovnic (2.11)

3 Slabá formulace, diskretizace

3.1 Časová diskretizace

K časové diskretizaci systému rovnic (2.11) využijeme dopřednou Eulerovu metodu. Její výhoda spočívá v jednoduché implementaci a výpočetní nenáročnosti jelikož jde o explicitní metodu. Na druhou stranu je to vykoupeno relativní nepřesností a nestabilitou (např. ve srovnání s implicitní Eulerovou metodou). Pro dostatečnou přesnost je třeba volit dostatečně malý časový krok.

$$\begin{aligned}y'(t) &= f(t, y(t)), y(t_0) = y_0 \\ y(t_{n+1}) &= y(t_n) + \Delta t f(t_n, y(t_n))\end{aligned}\tag{3.1}$$

Jedná se tedy o nejjednodušší lineární jednokrokovou metodu k řešení diferenciálních rovnic. Předpis lze odvodit z Taylorova rozvoje, kde navíc získáme i odhad lokální diskretizační chyby pro každý krok ($O(\Delta t^2)$), pro řešení problému na intervalu délky T je počet kroků proporcionální $1/\Delta t$ a tedy globální diskretizační chyba dopředné Eulerovy metody je řádu $O(\Delta t)$.

Pro přesnější časovou diskretizaci je také možno využít implicitní Eulerovu metodu nebo některou z Runge-Kuttových, případně lineárních vícekrokových metod [3, kap. 25].

3.2 Slabá formulace

Myšlenkou slabé formulace je zeslabení požadavků na hladkost řešení v původní formulaci, čímž připustíme více možných fyzikálních řešení.

Uvažujme počátečně-okrajovou úlohu z kapitoly 2.2, kde na členy s časovou derivací aplikujeme Eulerovu dopřednou metodu:

$$\begin{aligned}
\frac{1}{\Delta t} u^j - \nu \frac{\partial^2 u^j}{\partial x^2} + \frac{\partial u^j}{\partial x} + g \frac{\partial h^j}{\partial x} &= \frac{1}{\Delta t} u^{j-1}; j \geq 1, \\
\frac{1}{\Delta t} h^j + \frac{\partial (h^j u^j)}{\partial x} &= \frac{1}{\Delta t} h^{j-1}; j \geq 1, \\
u(x, 0) = u^0(x) = 0; \forall x \in \Omega = [0, l], \\
h(x, 0) = h^0(x) &= h_1; x < a \\
&= h_2; x \geq a, \\
u(0, t) = u(l, t) = 0 \forall t \in [0, T].
\end{aligned} \tag{3.2}$$

Přenásobíme první rovnici testovací funkcí $v \in V$ a druhou rovnici testovací funkcí $w \in W$, kde V a W jsou vhodné prostory testovacích funkcí. Otázkami existence jednoznačného řešení, tedy i správnou volbou prostorů testovacích funkcí V a W se zabývá např. [1]. Obě rovnice zintegrujeme přes interval $[0, l]$. Parciální derivace podle proměnné x budeme v dalším textu značit dolním indexem:

$$\begin{aligned}
\frac{1}{\Delta t} \int_{\Omega} u^j v \, dx - \nu \int_{\Omega} u_{xx}^j v \, dx + \int_{\Omega} u_x^j u^j v \, dx + g \int_{\Omega} h_x^j v \, dx &= \frac{1}{\Delta t} \int_{\Omega} u^{j-1} v \, dx; \\
\forall v \in V = H_0^1(\Omega), j \geq 1
\end{aligned} \tag{3.3}$$

$$\begin{aligned}
\frac{1}{\Delta t} \int_{\Omega} h^j w \, dx + \int_{\Omega} (h^j u^j)_x w \, dx &= \frac{1}{\Delta t} \int_{\Omega} h^{j-1} w; \\
\forall w \in W = H^1(\Omega), j \geq 1
\end{aligned}$$

Integrací per partes a dosazením okrajových podmínek pak dostaneme slabou formulaci úlohy

$$\begin{aligned}
u^j : u^j \in H_0^1(\Omega), u^j(0) = u^j(l) = 0, \\
\frac{1}{\Delta t} \int_{\Omega} u^j v \, dx + \nu \int_{\Omega} u_x^j v_x \, dx + \int_{\Omega} u_x^j u^j v \, dx + g \int_{\Omega} h^j v_x \, dx &= \frac{1}{\Delta t} \int_{\Omega} u^{j-1} v \, dx \\
\forall v \in H_0^1(\Omega), j \geq 1 \\
h^j : h^j \in H^1(\Omega), \\
\frac{1}{\Delta t} \int_{\Omega} h^j w \, dx + \int_{\Omega} (h^j u^j)_x w \, dx &= \frac{1}{\Delta t} \int_{\Omega} h^{j-1} w. \\
\forall w \in H^1(\Omega), j \geq 1
\end{aligned} \tag{3.4}$$

3.3 Prostorová diskretizace a MKP

Podstatou metody konečných prvků je nahrazení nekonečně-dimenzionálního prostoru testovacích funkcí (v našem případě V a W) prostorem konečně-dimenzionálním, u něhož bázi volíme tak, aby matice, které vzniknou diskretizací řešených úloh byly řídké, a tedy manipulace s nimi byla rychlá a efektivní.

Chceme tedy aproximovat prostor V konečně-dimenzionálním prostorem V_h a prostor W prostorem W_h tak, aby byla splněna LBB² podmínka. Proto budeme volit bázevské funkce prostoru V_h (resp. W_h) po částech kvadratické a lineární (resp. lineární a konstantní)³.

Uvažujme dělení T_h oblasti $\Omega = [0, l]$ na intervaly $e_i = [x_{i-1}, x_i]$, $i \in \{1, \dots, N\}$, $0 = x_0 < x_1 < \dots < x_{N-1} < x_N = l$. Na tomto dělení zdefinujeme prostory V_h a W_h takto:

$$V_{h,2} = \{v_h \in C(\Omega); v_h|_{e_i} \in P^2(e_i) \forall e_i \in T_h, v_h(0) = v_h(l) = 0\},$$

$$W_{h,1} = \{w_h \in C(\Omega); w_h|_{e_i} \in P^1(e_i) \forall e_i \in T_h\},$$

pro kvadraticko-lineární a

$$V_{h,1} = \{v_h \in C(\Omega); v_h|_{e_i} \in P^1(e_i) \forall e_i \in T_h, v_h(0) = v_h(l) = 0\},$$

$$W_{h,0} = \{w_h: \mathbb{R} \rightarrow \mathbb{R}; w_h|_{e_i} \in P^0(e_i) \forall e_i \in T_h\},$$

pro lineárně-konstantní aproximaci prostorů V a W .

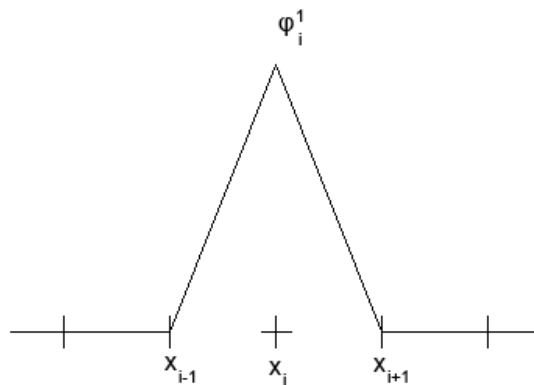
Nyní k volbě bázevských prostorů V_h a W_h . Pro po částech konstantní aproximaci stačí vzít jako bázi funkce φ_i^0 , které nabývají konstantní hodnoty na právě jednom intervalu e_i a na ostatních jsou nulové, tedy

$$\varphi_i^0(x) = \begin{cases} 1 & ; x \in e_i \\ 0 & ; x \notin e_i \end{cases}, i \in \{1, \dots, N\}. \quad (3.5)$$

Pro po částech lineární aproximaci je přirozené zdefinovat bázevskou funkci takto: $\varphi_i^1(x_j) = \delta_{i,j}$; $i, j \in \{0, \dots, N\}$, její znázornění je na obrázku 3.1.

2 LBB podmínka (Ladyženskaja-Babuška-Brezzi) je diskrétní obdobou inf-sup podmínky, viz [4]

3 Prostor $W_{h,0}$ sice není podprostor W , ale v kapitole 5.1 ukážeme, že i tato volba prostoru je postačující pro numerické řešení problému.



Obrázek 3.1: Po částech lineární bázová funkce φ_i^1

Na jednom intervalu e_i jsou tedy zdefinovány dvě lineární bázové funkce φ_{i-1}^1 a φ_i^1 :

$$\begin{aligned}\varphi_{i-1}^1(x) &= \frac{x_i - x}{|e_i|} ; x \in e_i \\ \varphi_i^1(x) &= \frac{x - x_{i-1}}{|e_i|} ; x \in e_i\end{aligned}\quad (3.6)$$

Bázové funkce pro kvadratickou aproximaci zdefinujeme obdobně jako u aproximace lineární. Tyto kvadratické funkce ale nejsou jednoznačně určeny na dělení oblasti T_h , na každém intervalu e_i potřebujeme jeden dodatečný uzel, tento uzel označíme $x_{i-1/2}$ a zdefinujeme takto

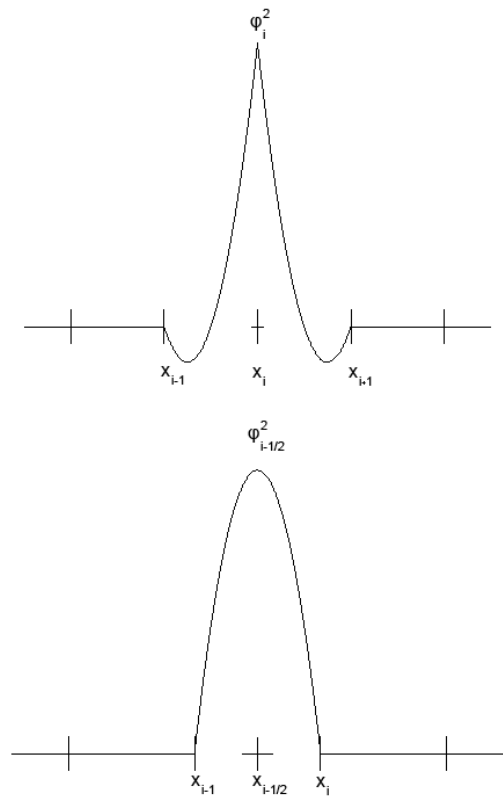
$$x_{i-1/2} = \frac{(x_i + x_{i-1})}{2} . \quad (3.7)$$

Kvadratické bázové funkce zdefinujeme stejně jako ty lineární, tedy $\varphi_i^2(x_j) = \delta_{i,j}$ s tím rozdílem, že množiny indexů i a j se nyní rozrostou o indexy nově definovaných uzlů: $i, j \in \{0, 1/2, \dots, N - 1/2, N\}$. Na každém intervalu e_i budou tedy nenulové nejvýše tři bázové funkce. Pro zpřehlednění zápisu ještě na každém intervalu e_i zavedeme lokální pomocnou proměnnou y : $y = (x - x_{i-1})/|e_i|$. Tato proměnná nabývá 0 na začátku a 1 na konci intervalu e_i . Touto pomocnou proměnnou už je pak vyjádření tří nenulových kvadratických

bázových funkcí jednoduché:

$$\begin{aligned}
 \varphi_{i-1}^2(x) &= (1-y(x))(1-2y(x)) ; x \in e_i \\
 \varphi_{i-1/2}^2(x) &= 4y(x)(1-y(x)) ; x \in e_i \\
 \varphi_i^2(x) &= -y(x)(1-2y(x)) ; x \in e_i
 \end{aligned}
 \tag{3.8}$$

Znázornění těchto funkcí je na obrázku 3.2. Funkce φ_{i-1}^2 je stejná jako φ_i^2 , jen posunutá o jeden element doleva.



Obrázek 3.2: Po částech kvadratické bázové funkce φ_i^2 , $\varphi_{i-1/2}^2$

Každý prvek prostoru V_h a W_h lze vyjádřit jako lineární kombinaci bázových funkcí daných prostorů, a proto hledáme-li řešení rovnic (3.4) na těchto prostorech, hledáme řešení ve tvaru

$$\begin{aligned}
u^j(x) &= \sum_i u_i^j \varphi_i^k(x) \\
h^j(x) &= \sum_i h_i^j \varphi_i^{k-1}(x) \quad ,
\end{aligned} \tag{3.9}$$

kde $k = 2$ pro kvadraticko-lineární, $k = 1$ pro lineárně-konstantní aproximaci, a rovnice (3.4) můžeme přepsat do tohoto tvaru:

$$\begin{aligned}
&u^j : u^j \in V_{h,k}, u^j(0) = u^j(l) = 0, \\
&\frac{1}{\Delta t} \int_{\Omega} u^j \varphi_i^k dx + \nu \int_{\Omega} u_x^j (\varphi_i^k)_x dx + \int_{\Omega} u_x^j u^j \varphi_i^k dx + g \int_{\Omega} h^j (\varphi_i^k)_x dx = \frac{1}{\Delta t} \int_{\Omega} u^{j-1} \varphi_i^k \\
&\quad \forall \varphi_i^k \in V_{h,k} \\
&h^j : h^j \in W_{h,k}, \\
&\frac{1}{\Delta t} \int_{\Omega} h^j \varphi_i^{k-1} dx + \int_{\Omega} h_x^j u^j \varphi_i^{k-1} dx + \int_{\Omega} h^j u_x^j \varphi_i^{k-1} dx = \frac{1}{\Delta t} \int_{\Omega} h^{j-1} \varphi_i^{k-1} \\
&\quad \forall \varphi_i^{k-1} \in W_{h,k}
\end{aligned} \tag{3.10}$$

Při vyčíslování jednotlivých integrálů rovnic se již projeví důležitost volby báze prostorů V_h a W_h . Výsledné matice totiž budou řídké, neboť integrály budou nenulové jen na malém počtu elementů (v závislosti na typu aproximace, u kvadraticko-lineární aproximace budou matice hustší).

Pro jednodušší a přehlednější zápis ještě přepíšeme systém rovnic (3.10) do maticové formy:

$$\begin{bmatrix} M_1^j + K^j + A^j(u) & B^j \\ C_1^j(h) + C_2^j(h) & M_2^j \end{bmatrix} \cdot \begin{bmatrix} \bar{u}^j \\ \bar{h}^j \end{bmatrix} = \begin{bmatrix} b_1^j \\ b_2^j \end{bmatrix}, \tag{3.11}$$

kde

$$\begin{aligned}
(M_1^j)_{i,m} &= \frac{1}{\Delta t} \int_{\Omega} \varphi_m^k \varphi_i^k dx, \\
(K^j)_{i,m} &= \nu \int_{\Omega} (\varphi_m^k)_x (\varphi_i^k)_x dx, \\
(A^j(u))_{i,m} &= - \int_{\Omega} (\varphi_m^k)_x u^j (\varphi_i^k)_x dx, \\
(B^j)_{i,m} &= g \int_{\Omega} \varphi_m^{k-1} (\varphi_i^k)_x dx, \\
(C_1^j(h))_{i,m} &= \int_{\Omega} h_x^j \varphi_m^k \varphi_i^{k-1} dx,
\end{aligned}$$

$$(C_2^j(h))_{i,m} = \int_{\Omega} h^j (\varphi_m^k)_x \varphi_i^{k-1} dx ,$$

$$(M_2^j)_{i,m} = \frac{1}{\Delta t} \int_{\Omega} \varphi_m^{k-1} \varphi_i^{k-1} dx ,$$

$$\bar{u}^j = [u_1^j, \dots, u_{N+1}^j]^T ,$$

$$\bar{h}^j = [h_1^j, \dots, h_N^j]^T ,$$

$$(b_1^j)_i = \frac{1}{\Delta t} \int_{\Omega} u^{j-1} \varphi_i^k dx ,$$

$$(b_2^j)_i = \frac{1}{\Delta t} \int_{\Omega} h^{j-1} \varphi_i^{k-1} dx .$$

Jak je vidět, jediné matice, které závisí na časovém kroku j jsou matice $A^j(u)$, $C_1^j(h)$, $C_2^j(h)$. Ostatní matice jsou konstantní, nebudeme tady u nich psát horní index j . Stačí je vygenerovat pouze jednou pro celou simulaci a ušetřit tím výpočetní čas. Vektory pravých stran b_1^j, b_2^j můžeme počítat za využití matic M_1, M_2 . Dosadíme z (3.9) do vyjádření vektorů pravých stran:

$$\begin{aligned} (b_1^j)_i &= \frac{1}{\Delta t} \int_{\Omega} \left(\sum_m u_m^{j-1} \varphi_m^k \right) \varphi_i^k dx = \sum_m u_m^{j-1} \frac{1}{\Delta t} \int_{\Omega} \varphi_m^k \varphi_i^k dx , \\ b_1^j &= M_1 u^{j-1} \end{aligned} \quad (3.12)$$

$$\begin{aligned} (b_2^j)_i &= \frac{1}{\Delta t} \int_{\Omega} \left(\sum_m h_m^{j-1} \varphi_m^{k-1} \right) \varphi_i^{k-1} dx = \sum_m h_m^{j-1} \frac{1}{\Delta t} \int_{\Omega} \varphi_m^{k-1} \varphi_i^{k-1} dx . \\ b_2^j &= M_2 h^{j-1} \end{aligned}$$

3.4 Vyčíslení MKP matic

Matice ve vyjádření (3.11) lze sestavit po uzlech. Tento přístup se nabízí jako první, ovšem jeho nevýhoda spočívá v tom, že při počítání hodnot matic v určitém uzlu potřebujeme znát i uzly s ním sousedící. U jedno-dimenzionálního modelu to není problém, ten nastává až při výpočtu ve více dimenzích. Proto využijeme přístupu, který sestavuje matice přes jednotlivé elementy. Tento přístup je také přehlednější a lépe algoritimizovatelný. Tento přístup popíšeme

při sestavování matice K pro lineární prvky. U ostatních matic je postup obdobný.

Pro sestavení matice K potřebujeme napočítat integrály $(K)_{i,m} = \nu \int_{\Omega} (\varphi_m^2)_x (\varphi_i^2)_x dx$, které si dále můžeme rozepsat přes jednotlivé intervaly e_l . Jelikož ale námi definované báze funkce jsou nenulové pouze na určitých intervalech, můžeme spočítat pouze lokální matice elementů a z nich pak sestavit globální matici K . Na každém intervalu e_l jsou nenulové pouze dvě lineární báze funkce, a tedy lokální matice K_l bude 2×2 :

$$K_l = \nu \begin{bmatrix} \int_{e_l} (\varphi_{l-1}^1)_x (\varphi_{l-1}^1)_x dx & \int_{e_l} (\varphi_{l-1}^1)_x (\varphi_l^1)_x dx \\ \int_{e_l} (\varphi_l^1)_x (\varphi_{l-1}^1)_x dx & \int_{e_l} (\varphi_l^1)_x (\varphi_l^1)_x dx \end{bmatrix}, \quad (3.13)$$

Dále zadefinujeme matici B_l derivací báze funkcí:

$$B_l = [(\varphi_{l-1}^1)_x \quad (\varphi_l^1)_x]. \quad (3.14)$$

Pomocí této matice můžeme přepsat matici elementů (3.13) do této podoby:

$$K_l = \nu \int_{e_l} B_l^T B_l dx. \quad (3.15)$$

U jedno-dimenzionálního modelu je již napočítání integrálů podle (3.15) jednoduché, ve více dimenzích je však výhodné tyto integrály počítat na referenčním elementu E . Pro jednoduchost budeme krajní body elementu e_l na kterém chceme spočítat lokální matici značit $x^{(1)}, x^{(2)}$. Na zobrazení $F: \hat{x} \rightarrow x$, které bude zobrazovat referenční element na e_l budeme klást tyto nároky: $F(0) = x^{(1)}, F(1) = x^{(2)}$, F je afinní zobrazení. Z těchto požadavků dostaneme předpis pro zobrazení F :

$$F(\hat{x}) = x^{(1)} + \hat{x}(x^{(2)} - x^{(1)}). \quad (3.16)$$

Nyní můžeme svázat předpisy pro báze funkce φ_i a $\hat{\varphi}_i$ pomocí zobrazení F : $\varphi_i \circ F = \hat{\varphi}_i$, a derivaci báze funkce na referenčním elementu spočítáme jako derivaci složené funkce:

$$\hat{B} = \begin{bmatrix} \frac{\partial \hat{\varphi}_1}{\partial \hat{x}} & \frac{\partial \hat{\varphi}_2}{\partial \hat{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_2}{\partial x} \end{bmatrix} \frac{\partial F}{\partial \hat{x}} = J^T B, \quad J^T = \frac{\partial F}{\partial \hat{x}}, \quad (3.17)$$

kde J je jakobián zobrazení F^t . Ze vzorce (3.17) už získáme předpis pro výpočet matice B_l pro

4 Ve vzorci (3.17) je transpozice jakobiánu. U jedno-dimenzionálního problému je jakobián pouze skalár, ovšem u problému ve více dimenzích jde o matici a tam už je transpozice zapotřebí.

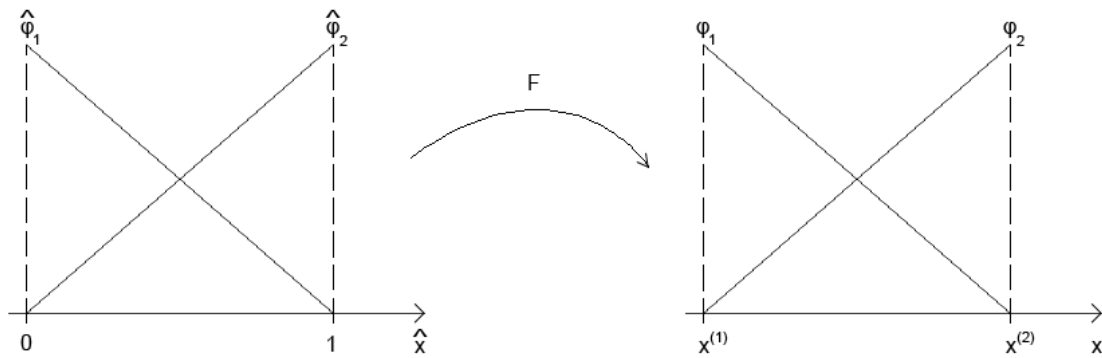
vzorec (3.15), potřebujeme ale ještě matici \hat{B} . Na referenčním elementu E jsou definovány dvě lineární báze funkce

$$\begin{aligned}\hat{\varphi}_1(\hat{x}) &= a_1 + b_1 \hat{x} \\ \hat{\varphi}_2(\hat{x}) &= a_2 + b_2 \hat{x}\end{aligned}\quad (3.18)$$

kteří jsou jednoznačně určeny funkčními hodnotami v krajních bodech referenčního elementu \hat{x}_1, \hat{x}_2 :

$$\begin{aligned}\hat{\varphi}_1(\hat{x}_1) &= 1, \hat{\varphi}_1(\hat{x}_2) = 0 \\ \hat{\varphi}_2(\hat{x}_1) &= 0, \hat{\varphi}_2(\hat{x}_2) = 1\end{aligned}\quad (3.19)$$

Z rovnic (3.18) a (3.19) pak pro každou bázeovou funkci můžeme sestavit soustavu, která určuje neznámé koeficienty a_i, b_i .



Obrázek 3.3: Zobrazení F zobrazující referenční element na element e_i

$$\begin{aligned}\hat{\varphi}_1(\hat{x}) &: \begin{bmatrix} 1 & \hat{x}_1 \\ 1 & \hat{x}_2 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ \hat{\varphi}_2(\hat{x}) &: \begin{bmatrix} 1 & \hat{x}_1 \\ 1 & \hat{x}_2 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.\end{aligned}\quad (3.20)$$

Pro matici \hat{B} potřebujeme derivaci těchto funkcí, zajímají nás tedy jen koeficienty b_1, b_2 . Tyto koeficienty vypočítáme pomocí Cramerova pravidla.

$$D \equiv \begin{vmatrix} 1 & \hat{x}_1 \\ 1 & \hat{x}_2 \end{vmatrix},$$

$$b_1 = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} D^{-1} = \tilde{b}_1 D^{-1}, \quad (3.21)$$

$$b_2 = \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix} D^{-1} = \tilde{b}_2 D^{-1}.$$

Nakonec zadefinujeme matici $\tilde{B} = [\tilde{b}_1, \tilde{b}_2]$. Dle (3.17), (3.18) a (3.21) platí

$$\tilde{B} D^{-1} = \hat{B} = J^T B,$$

a tedy

$$B = J^{-T} \tilde{B} D^{-1} \quad (3.22)$$

dosazením (3.22) do (3.15) tak dostaneme

$$B = J^{-T} \tilde{B} D^{-1}$$

$$K_l = \nu \int_{e_l} (J^{-T} \tilde{B} D^{-1})^T J^{-T} \tilde{B} D^{-1} dx = \nu \int_{e_l} \tilde{B}^T J^{-1} J^{-T} \tilde{B} D^{-2} dx \quad (3.23)$$

$$= \tilde{B}^T J^{-1} J^{-T} \tilde{B} |D|^{-1}$$

Nyní již můžeme sestavit globální matici K . K tomu potřebujeme lokální matice K_l , zobrazení σ_l , které které zobrazuje lokální indexy uzlů na globální a seznam uzlů s předepsanou Dirichletovou okrajovou podmínkou I_D . Při použití lineárních prvků v jedno-dimenzionálním modelu je zobrazení σ_l definováno následovně:

$$\sigma_l: \{1, 2\} \rightarrow \{0, \dots, N\}$$

$$\sigma_l(1) = l-1$$

$$\sigma_l(2) = l$$

Algoritmus který bude sestavovat globální matici K pak bude vypadat následovně:

```
funkce Assemble_K
01   for l = 1 ... N
02       for j = 1 ... m
03           for k = 1 ... m
04               if (j==k) ∨ (σ1(j) ∉ ID & σ1(k) ∉ ID)
```

```
05             (K) $\sigma_1(j), \sigma_1(k)$  += (K1)j,k
06         end if
07     end for
08 end for
09 end for
```

Konec algoritmu, v proměnné K je uložen výsledek

4 Numerické řešení

Jak je vidět z maticové formulace (3.11), potřebujeme v každém časovém kroku j řešit soustavu algebraických rovnic pro neznámé \bar{u}^j, \bar{h}^j . Bohužel je tato soustava nelineární, a proto je třeba použít nějakou speciální metodu pro řešení nelineárních rovnic. Touto metodou bude Newtonova metoda.

4.1 Newtonova metoda

Nechť je dána (nelineární) rovnice

$$F(x) = 0; F: \mathbb{R} \rightarrow \mathbb{R}. \quad (4.1)$$

Ekvivalentními úpravami ji můžeme převést na tvar $x = x + \lambda(x) \cdot F(x)$.

Hledání kořene rovnice (4.1) je tedy ekvivalentní k hledání pevného bodu zobrazení

$$G(x) = x + \lambda(x) \cdot F(x), \quad (4.2)$$

a iterační předpis bude tohoto tvaru:

$$x_{n+1} = G(x_n) = x_n + \lambda(x_n) \cdot F(x_n). \quad (4.3)$$

Po Newtonově metodě požadujeme, aby byla řádu dvě, první nenulová derivace $G(x)$ musí být tedy také řádu dvě⁵. Derivováním pravé strany rovnice (4.2) a dosazením řešení rovnice (4.1) dostaneme podmínku na koeficient $\lambda(x)$:

- 5 Nechť $\{x_n\}$ je posloupnost aproximací kořene rovnice (4.1) a x^* je kořen této rovnice. Řekneme, že iterační metoda je řádu Γ tehdy a jen tehdy, platí-li následující nerovnost:

$$\exists C > 0: |x^* - x_{n+1}| \leq C |x^* - x_n|^\Gamma \text{ pro } n = 0, 1, \dots$$

Označme chybu aproximace $e_{n+1} = x_{n+1} - x^*$ a rozviňme ji Taylorovým rozvojem v okolí bodu x^* :

$$e_{n+1} = x_{n+1} - x^* = G(x_n) - G(x^*) = -G'(x^*) \cdot (x_n - x^*) + \sum_{k=2}^{\infty} \frac{G^{(k)}(x^*)}{k!} \cdot (x_n - x^*)^k.$$

Využitím Lagrangeova tvaru zbytku pak chybu aproximace můžeme vyjádřit takto:

$$e_{n+1} = \sum_{k=1}^m \frac{G^{(k)}(x^*)}{k!} \cdot e_n^k + \frac{G^{(m+1)}(\xi)}{(m+1)!} \cdot e_n^{m+1}; \xi \in (\min(x_n, x^*), \max(x_n, x^*)).$$

Pokud jsou derivace $G(x)$ v bodě x^* rovny nule až do řádu m , závisí chyba aproximace e_{n+1} pouze na $(m+1)$ -ní derivaci $G(x)$ v bodě x^* . Řád metody tedy závisí na řádu první nenulové derivace $G(x)$ v bodě x^* .

$$\lambda(x) = -\frac{1}{F'(x)}. \quad (4.4)$$

Dosazením do rovnice (4.2) tedy získáváme iterační předpis pro hledání pevného bodu zobrazení (4.2) těmito iteracemi:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}. \quad (4.5)$$

Uvažujeme-li soustavu rovnic

$$\vec{F}(x) = \vec{0}; F: \mathbb{R}^m \rightarrow \mathbb{R}^n, \quad (4.6)$$

můžeme iterační schéma (4.5) přepsat do tohoto tvaru:

$$x^{k+1} = x^k - (F'(x^k))^{-1} \cdot F(x^k), \quad (4.7)$$

kde $F'(x^k)$ je jakobián matice $F(x^k)$ ⁶. V každé iteraci je tedy třeba počítat inverzi matice, což je zvláště při větších soustavách výpočetně velmi náročné, iterace se proto počítají takto:

$$F'(x^k) \Delta x^k = -F(x^k), \quad (4.8)$$

$$x^{k+1} = x^k + \Delta x^k. \quad (4.9)$$

V rovnici (4.8) řešíme lineární soustavu rovnic, což je v poměru k počítání inverze matice výpočetně levná záležitost. Další vylepšení Newtonovy metody spočívá v pozorování, že směr Δx^k může být správný, ale touto iterací "přestřelíme" řešení a následná iterace tak bude zatížena větší chybou jak je ilustrováno na obrázku 4.1. Proto do metody zavedeme tlumicí parametr α s počáteční hodnotou $\alpha=1$.

$$x^{k+1} = x^k + \alpha \Delta x^k. \quad (4.10)$$

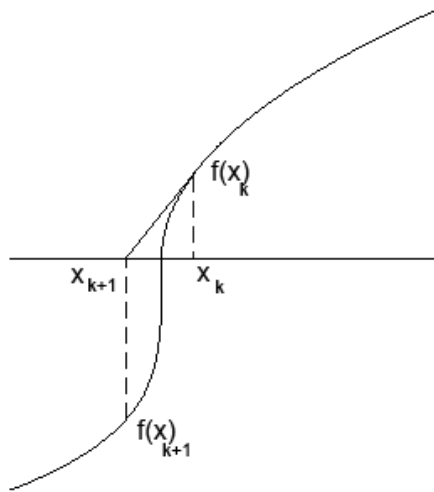
Vektor Δx^k nám udává, ve kterém směru hodnota $F(x^k)$ klesá. Pokud by se stalo, že hodnota v následujícím kroku $F(x^{k+1})$ je vyšší, stačí zmenšit tlumicí parametr a provést novou iteraci.

6 Funkce (4.6) je reprezentována n reálnými funkcemi m proměnných. Označíme-li každou takovouto funkci

$$F_i(x) = 0; F_i: \mathbb{R}^m \rightarrow \mathbb{R}; i \in \{1, 2, \dots, n\},$$

a existují-li všechny parciální derivace těchto funkcí, můžeme jakobián $F'(x)$ definovat jako matici, pro jejíž prvky platí následující:

$$F'_{ij}(x) = \frac{\partial F_i(x)}{\partial x_j}; i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}.$$



Obrázek 4.1: Ilustrace k nutnosti bisekce v
Newtonově metodě

Symbolický zápis Newtonovy metody:

funkce Newton(x_0, eps)

```

01   x = x0;
02   fx = f(x);
03   norm1 = norm2 = norm(fx);
04   while (norm(dx)/norm1 >= eps)
05       alfa = 1;
06       fx = f(x);
07       dfx = f'(x);           (jakobián funkce f(x))
08       dx = dfx\fx;          (řešení lineární soustavy)
09   for i = 1 ...
10       x1 = x + alfa·dx;
11       norm3 = norm(f(x1));
12       if (norm(fx1) < norm2)

```

```

13             break;
14         end if
15         alfa = alfa/2;
16     end for
17     x = x1;
18     norm2 = norm3;
19 end while

```

Konec algoritmu, v proměnné x je uložen výsledek

Bisekce v Newtonově metodě probíhá na řádcích 09-16, x_0 je počáteční aproximace a eps je požadovaná přesnost řešení. Ukončovací kritérium Newtonovy metody může být také nastaveno tak, aby celkový počet iterací nepřesáhl určitou hodnotu, také je možno omezit maximální možný počet bisekcí v jedné iteraci.

Abychom mohli na soustavu (3.11) použít Newtonovu metodu, musíme ji převést na tvar ekvivalentní k (4.6):

$$F(x) = \begin{bmatrix} M_1 + K + A^j(u) & B \\ C_1^j(h) + C_2^j(h) & M_2 \end{bmatrix} \cdot \begin{bmatrix} \bar{u}^j \\ \bar{h}^j \end{bmatrix} - \begin{bmatrix} b_1^j \\ b_2^j \end{bmatrix} = 0, x = \begin{bmatrix} \bar{u}^j \\ \bar{h}^j \end{bmatrix}. \quad (4.11)$$

V každém kroku Newtonovy metody pak musíme řešit lineární soustavu (4.8). Tuto soustavu budeme řešit metodou GMRES.

4.2 GMRES

GMRES (z anglického generalized minimal residual method) je iterační metoda pro řešení soustav lineárních rovnic. Metoda aproximuje řešení v Krylovových prostorech generovaných maticí soustavy a zároveň konstruuje jejich bázi v Arnoldiho procesu.

funkce GMRES (x_0)

```

01     r0 = A · x0 - b;

```

```

02      $\beta = \text{norm}(r_0);$ 
03      $v_1 = r_0/\beta;$ 
04     for  $j = 1 \dots n$ 
05          $w_j = A \cdot v_j;$ 
06         for  $i = 1 \dots j$ 
07              $h_{i,j} = \langle w_j, v_i \rangle;$ 
08              $w_j = w_j - h_{i,j} v_i;$ 
09         end for
10          $h_{j+1,j} = \text{norm}(w_j);$ 
11          $v_{j+1} = w_j/h_{j+1,j};$ 
12     end for
13      $y_m = \text{argmin} \{ \beta e_1 - H_m \cdot y_m \};$ 
14      $x_m = x_0 - V_m \cdot y_m;$ 

```

Konec algoritmu, v proměnné x_m je uložen výsledek

Algoritmus generuje v každém kroku nový bázeový vektor v_{j+1} zvětšujících se Krylovových prostorů. Tyto vektory se ukládají do matice V_m , kde m je počet kroků, ve kterém metoda konverguje k řešení s požadovanou přesností. U rozsáhlých soustav tak vyvstává problém s paměťovou náročností, neboť pro konstrukci přibližného řešení potřebujeme znát všechny vektory v_i .

Jedním z možných řešení paměťové náročnosti metody GMRES je tzv. restartovaná metoda GMRES(R). Její základní myšlenka spočívá v tom, že po určitém počtu iterací R spočteme přibližné řešení x_R a algoritmus restartujeme s počáteční aproximací $x_0 = x_R$. Zde ovšem narážíme na další problém – jak volit parametr R pro restart algoritmu. Pro rostoucí R bychom očekávali, že rychlost konvergence se bude zvyšovat, bohužel obecně to není pravda, existují soustavy kdy GMRES(k) pro konverguje v určitém počtu kroků a GMRES(n) nekonverguje pro $n \geq k$. Toto chování také velmi ovlivňuje volba počáteční aproximace x_0 .

funkce GMRES(x_0, R)

```

01      $r_0 = A \cdot x_0 - b;$ 

```

```

02    $\beta = \text{norm}(r_0);$ 
03    $v_1 = r_0/\beta;$ 
04   for j = 1 ... R
05        $w_j = A \cdot v_j;$ 
06       for i = 1 ... j
07            $h_{i,j} = \langle w_j, v_i \rangle;$ 
08            $w_j = w_j - h_{i,j} v_i;$ 
09       end for
10        $h_{j+1,j} = \text{norm}(w_j);$ 
11        $v_{j+1} = w_j/h_{j+1,j};$ 
12   end for
13    $y_R = \text{argmin} \{ \beta e_1 - H_R \cdot y_R \};$ 
14    $x_R = x_0 - V_R \cdot y_R;$ 
15    $x_0 = x_R;$ 
16   GoTo 01

```

Konec algoritmu, v proměnné x_R je uložen výsledek

Jako u každé iterační metody, i zde můžeme použít předpodmínění, které významným způsobem snižuje počet iterací. Předpodmínění spočívá v tom, že matici soustavy nahradíme maticí jí podobnou, která je ovšem lépe podmíněná, a iterační metody, kterých rychlost konvergence je závislá na číslu podmíněnosti soustavy, tedy konvergují k přibližnému řešení v menším počtu iterací.

Podrobnější popis metody GMRES a různých předpodmiňovačů lze nalézt v [6].

5 Numerické simulace

Tato kapitola se bude věnovat numerickému řešení rovnic (2.11) resp. jejich MKP formulaci (3.10). Z důvodu přítomnosti advektivního členu v těchto rovnicích nelze volit časový diskretizační krok Δt a prostorový diskretizační krok Δx libovolně. To vyjadřuje CFL podmínka:

$$u \frac{\Delta t}{\Delta x} < C, C \in \mathbb{R}^+ .$$

Tato podmínka má i celkem zřejmý fyzikální smysl: časová diskretizace musí být volena tak, aby vlna o rychlosti u za dobu Δt nemohla opustit element o délce Δx . V kapitole 3.1 byla pro časovou diskretizaci zvolena dopředná Eulerova metoda. Pro dostatečnou přesnost této metody je třeba volit dostatečně malý krok, což opět koresponduje s CFL podmínkou. Vliv volby délky časového diskretizačního kroku pak bude patrný i v rychlosti konvergence metody GMRES.

V následujících simulacích bude počítáno s těmito parametry:

$$l = 5 \text{ m} ,$$

$$g = 9.80665 \text{ m s}^{-2} \text{ (střední hodnota gravitačního zrychlení),}$$

$$\nu = 10^{-6} \text{ m}^2 \text{ s}^{-1} \text{ (odpovídá viskozitě vody při 20°C),}$$

$$h_1 = 1.5 \text{ m} ,$$

$$h_2 = 1.0 \text{ m} ,$$

$$a = 1.2 \text{ m} .$$

Hodnoty řešení v čase $t = 0 \text{ s}$ jsou známy z počáteční podmínky. Ty také tvoří počáteční iteraci řešení v čase $t = \Delta t$, která vstupuje do Newtonovy metody.

Pro simulace s lineárně-konstantní i kvadraticko-lineární aproximací budeme počítat s následujícími časovými a prostorovými diskretizačními kroky:

$$\Delta t \in \{0.01 \text{ s}, 0.02 \text{ s}, 0.04 \text{ s}, 0.08 \text{ s}, 0.16 \text{ s}\} ,$$

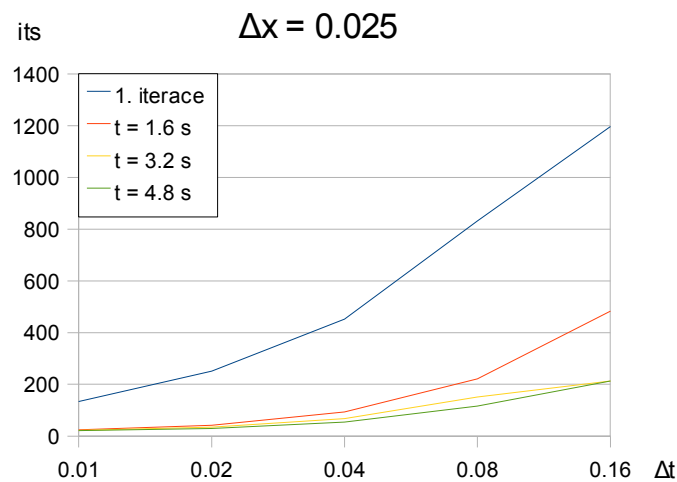
$$\Delta x \in \{0.025 \text{ m}, 0.05 \text{ m}, 0.10 \text{ m}, 0.20 \text{ m}\} .$$

5.1 Simulace s lineárně-konstantními prvky

V následujících grafech je pro různé prostorové diskretizační kroky porovnán počet iterací metody GMRES v jednotlivých časových krocích při použití lineárně-konstantních prvků. Jelikož je metoda GMRES volána při každé Newtonově iteraci, je počet iterací v grafech odpovídající jednomu časovému kroku součtem počtu iterací ve všech bžích metody GMRES

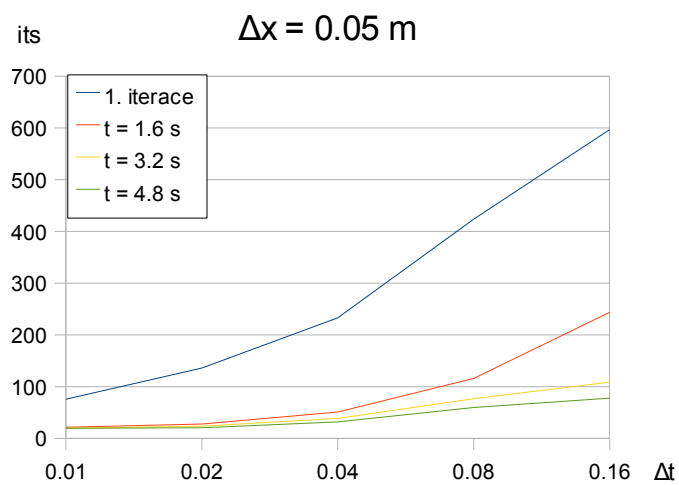
v rámci celého běhu Newtonovy metody.⁷

Z grafů je patrné, že nejvíce GMRES iterací bylo zapotřebí u první časové iterace, což je zřejmě způsobeno počátečním skokem ve výšce hladiny h . Nejméně GMRES iterací bylo zapotřebí u nejjemnější časové diskretizace. To je na druhou stranu vykoupeno tím, že pro simulaci do určitého času T je u jemnější diskretizace zapotřebí provést více časových iterací. Z grafů je dále vidět, že pro hrubší časovou diskretizaci poměrně významně roste počet GMRES iterací, což souvisí právě s již zmiňovanou CFL podmínkou, kdy se matice soustavy stává špatně podmíněnou.

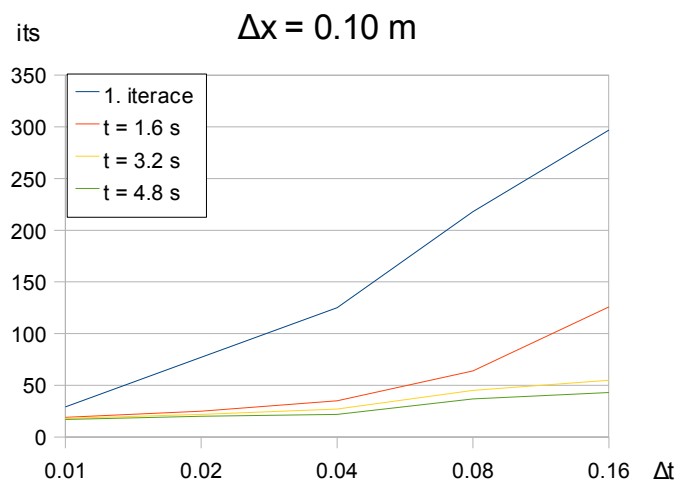


Graf 1: Počet iterací pro lineárně-konstantní prvky v závislosti na časové diskretizaci pro $\Delta x = 0.025$ m

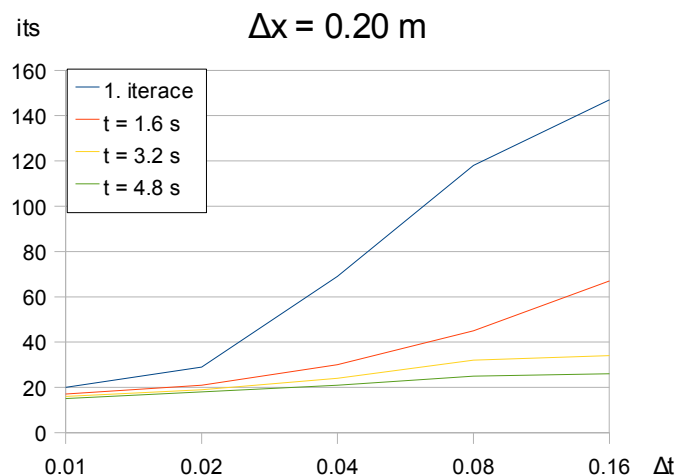
⁷ Pokud pro přechod z časového kroku j do $j+1$ bylo zapotřebí tří iterací Newtonovy metody a v těchto iteracích proběhly podtupně 7, 5 a 2 iterace GMRES, je celkový počet GMRES iterací v rámci jednoho časového kroku 14.



Graf 2: Počet iterací pro lineárně-konstantní prvky v závislosti na časové diskretizaci pro $\Delta x = 0.05 \text{ m}$



Graf 3: Počet iterací pro lineárně-konstantní prvky v závislosti na časové diskretizaci pro $\Delta x = 0.10 \text{ m}$



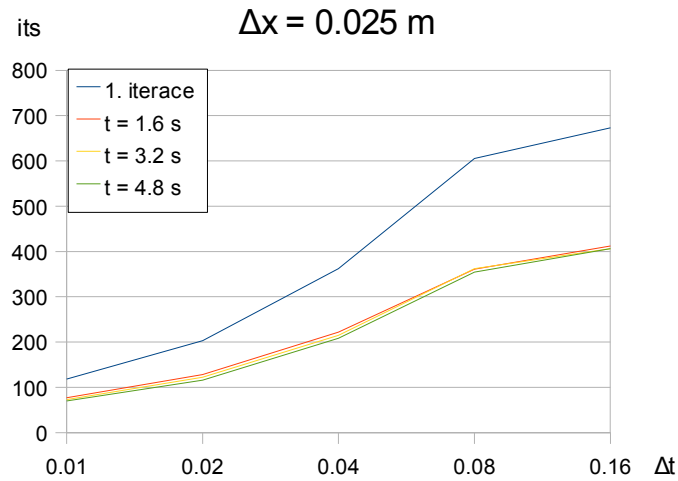
Graf 4: Počet iterací pro lineárně-konstantní prvky v závislosti na časové diskretizaci pro $\Delta x = 0.20 \text{ m}$

5.2 Simulace s kvadraticko-lineárními prvky

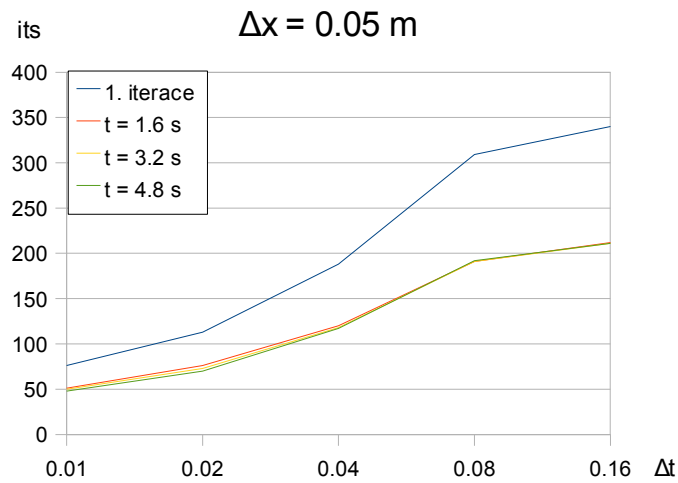
V následujících grafech je pro různé prostorové diskretizační kroky porovnán počet iterací metody GMRES v jednotlivých časových krocích při použití kvadraticko-lineárních prvků.

Podobně jako u lineárně-konstantních prvků, nejvíce GMRES iterací bylo zapotřebí u první časové iterace, nejméně iterací bylo zapotřebí u nejjemnější časové diskretizace.

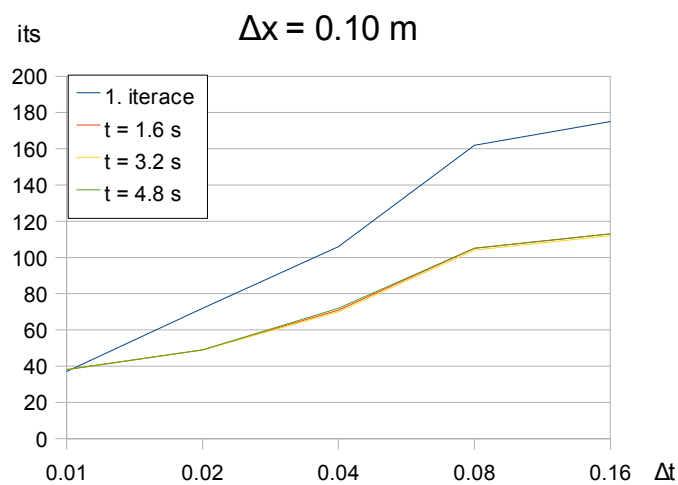
Pro stejnou časovou i prostorovou diskretizaci bylo u kvadraticko-lineárních prvků zapotřebí méně GMRES iterací, na druhou stranu každá GMRES iterace byla výpočetně náročnější, neboť MKP matice jsou pro prvky vyššího stupně více zaplněné.



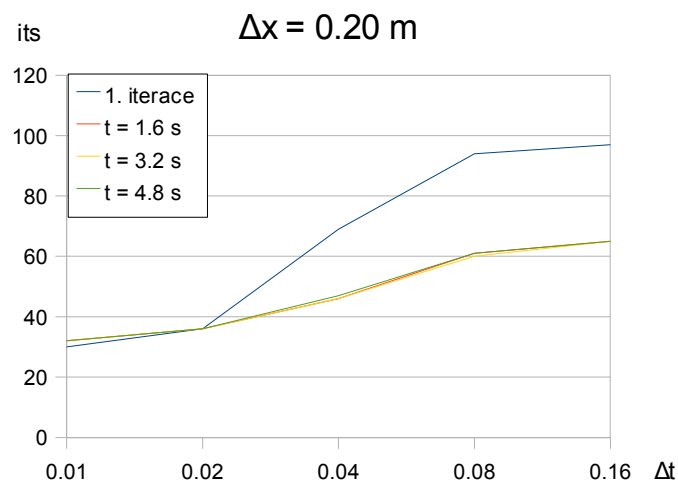
Graf 5: Počet iterací pro kvadraticko-lineární prvky v závislosti na časové diskretizaci pro $\Delta x = 0.025 \text{ m}$



Graf 6: Počet iterací pro kvadraticko-lineární prvky v závislosti na časové diskretizaci pro $\Delta x = 0.05 \text{ m}$



Graf 7: Počet iterací pro kvadraticko-lineární prvky v závislosti na časové diskretizaci pro $\Delta x = 0.10 \text{ m}$



Graf 8: Počet iterací pro kvadraticko-lineární prvky v závislosti na časové diskretizaci pro $\Delta x = 0.20 \text{ m}$

6 Závěr

Cílem této práce bylo numerické řešení rovnic povrchového proudění metodou konečných prvků. Proto byly ve druhé kapitole odvozeny Saint-Venantovy rovnice a ve třetí kapitole jejich slabá formulace, diskretizace a MKP formulace. Následující kapitola byla věnována numerickým metodám použitým k řešení vzniklých algebraických rovnic. Poslední kapitola byla věnována numerickým simulacím s různou volbou konečných prvků.

Zefektivnění numerického řešení by bylo možno dosáhnout tím, že by se lineární soustavy, které vznikají v rámci Newtonovy metody, řešily pomocí metody GMRES s adaptivní přesností - čím blíže bychom byli řešení v rámci Newtonovy metody, tím přesněji bychom museli řešit lineární soustavy pomocí GMRES. Tím bychom ušetřili jistý počet GMRES iterací v každém kroku Newtonovy metody.

Pro řešení Saint-Venantových je možné použít také nepojitou Galerkinovu metodu (DGFEM - Discontinuous Galerkin Finite Element Method). Její výhoda spočívá v tom, že neklade žádné nároky na spojitost řešení mezi jednotlivými elementy, jako je tomu u klasické metody konečných prvků, a tedy lépe postihuje řešení s nespojitostmi nebo velkými gradienty [1].

7 Literatura

- [1] FEISTAUER, Miloslav, FELCMAN, Jiří, STRAŠKRABA, Ivan. *Mathematical and Computational Methods for Compressible Flow*. New York : Oxford University Press, 2003. xiii, 535 s. ISBN 0-19-850588-4.
- [2] REKTORYS, Karel, et al. *Přehled užití matematiky I*. 7. vyd. Praha : Prometheus , 2000. xxxii, 720 s. ISBN 80-7196-180-9.
- [3] REKTORYS, Karel, et al. *Přehled užití matematiky II*. 7. vyd. Praha : Prometheus , 2000. xxxii, 874 s. ISBN 80-7196-181-7
- [4] FEISTAUER, Miloslav. *Theory and Numerics for Problems of Fluid Dynamics* [online]. 2006 [cit. 2009-04-29].
Dostupný z WWW: <<http://www.karlin.mff.cuni.cz/~feist/MMFD.pdf>>.
- [5] SOKOL, Vojtěch. *Simulace 1-rozměrné vlny při protržení přehrady*. 2007 VŠB - Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. Bakalářská práce.
Dostupný z WWW: <<http://am.vsb.cz/theses/bakalari/2007/pdfs/sokol.pdf>>
- [6] SAAD, Yousef. *Iterative methods for sparse linear systems*. 1. vyd. Boston : PWS Publishing Company, 1995. xvi, 447 s. ISBN 0-534-94776-X.
Dostupný z WWW: <http://www-users.cs.umn.edu/~saad/PS/all_pdf.zip>.

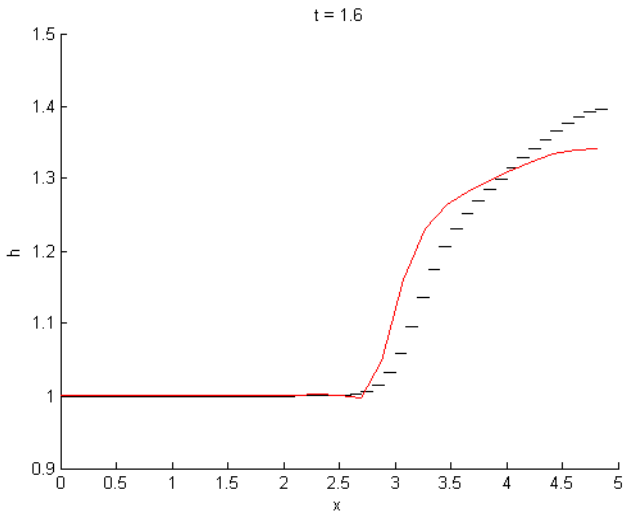
8 Přílohy

K diplomové práci je přiloženo CD, které obsahuje

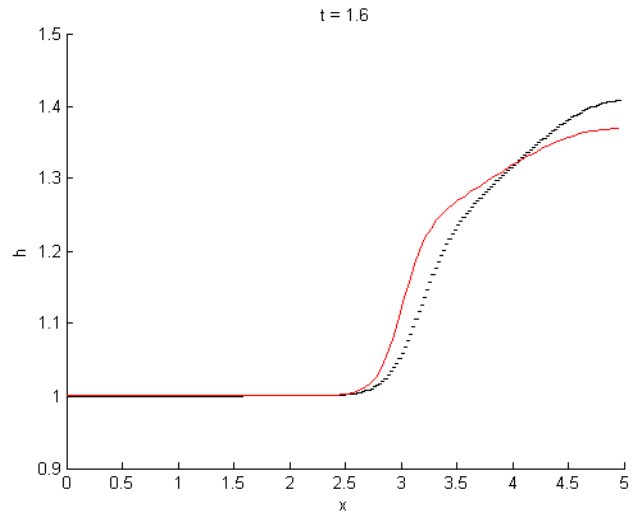
- tuto práci ve formátu .pdf,
- zdrojové kódy programů pro Matlab,
- výstupy numerických simulací.

Simulace v čase $t = 1.6$ s

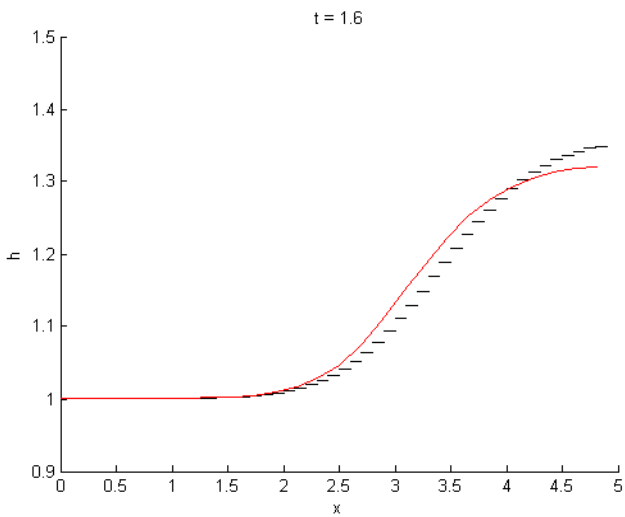
Porovnání výšek h pro konstantní (černě) a lineární (červeně) v čase $t = 1.6$ s.



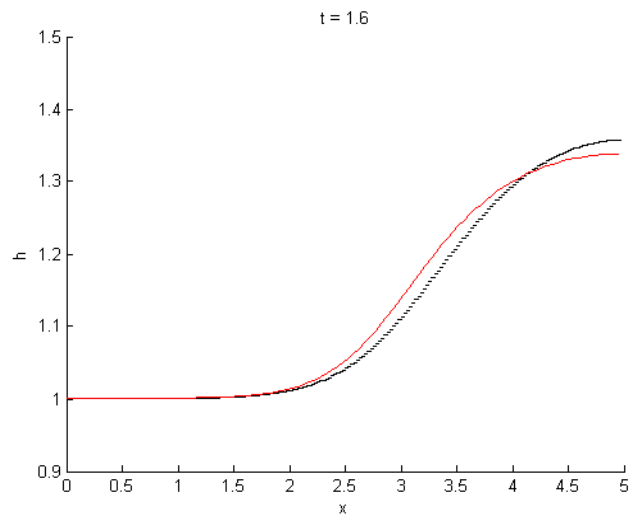
$\Delta x = 0.10$ m, $\Delta t = 0.01$ s



$\Delta x = 0.025$ m, $\Delta t = 0.01$ s



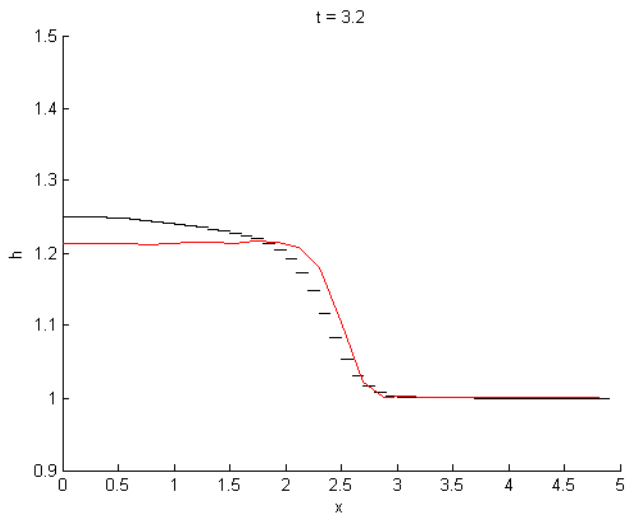
$\Delta x = 0.10$ m, $\Delta t = 0.04$ s



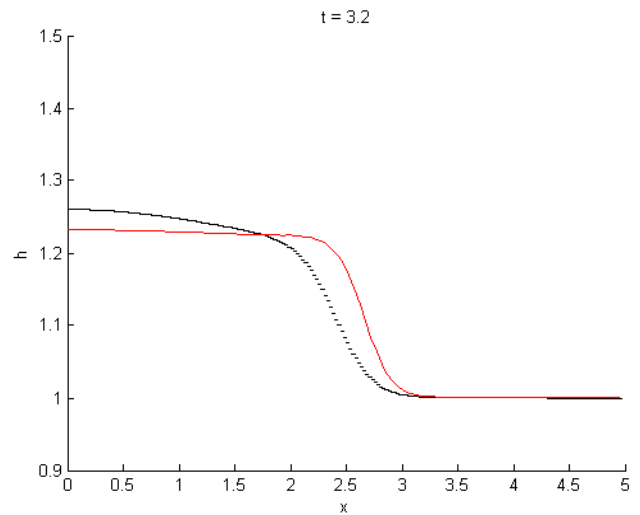
$\Delta x = 0.025$ m, $\Delta t = 0.04$ s

Simulace v čase $t = 3.2$ s

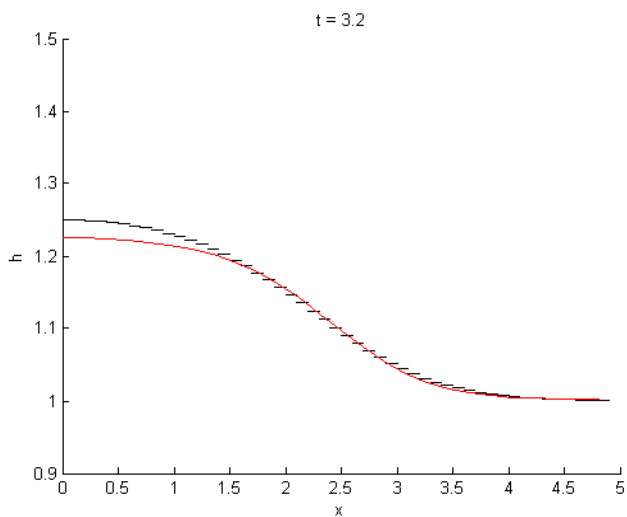
Porovnání výšek h pro konstantní (černě) a lineární (červeně) v čase $t = 3.2$ s.



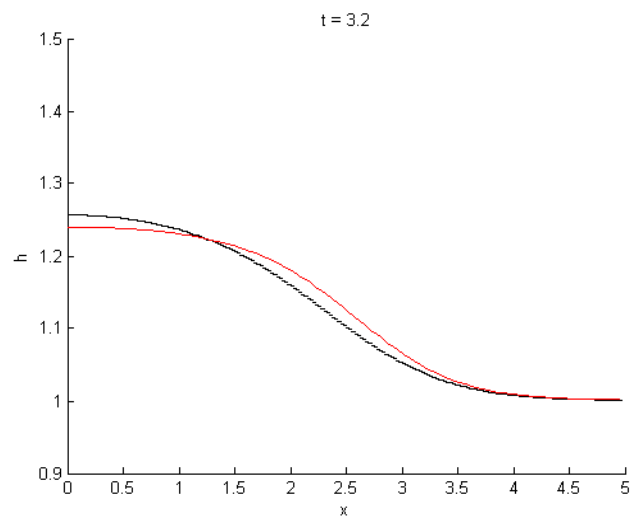
$\Delta x = 0.10$ m, $\Delta t = 0.01$ s



$\Delta x = 0.025$ m, $\Delta t = 0.01$ s



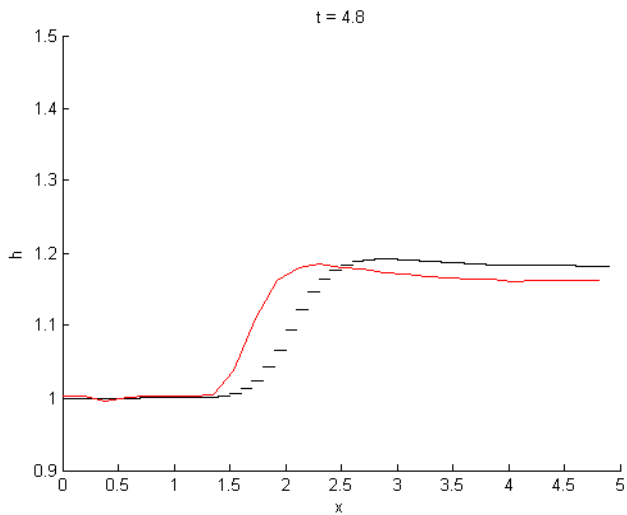
$\Delta x = 0.10$ m, $\Delta t = 0.04$ s



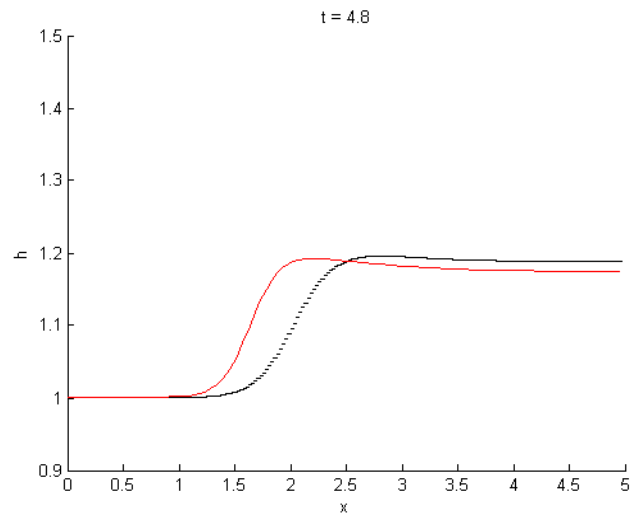
$\Delta x = 0.025$ m, $\Delta t = 0.04$ s

Simulace v čase $t = 4.8$ s

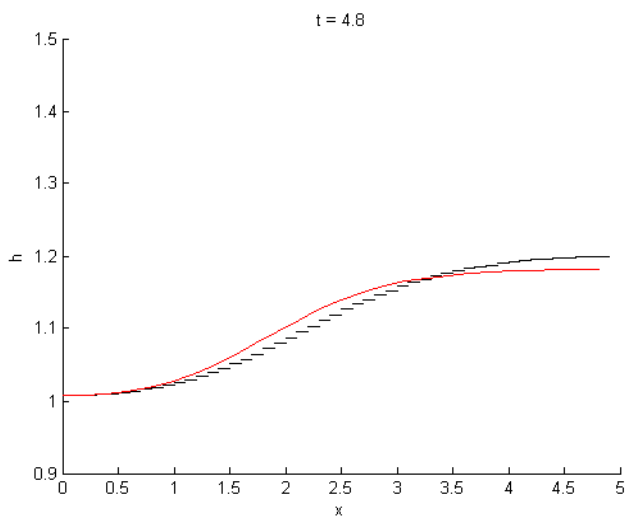
Porovnání výšek h pro konstantní (černě) a lineární (červeně) v čase $t = 4.8$ s.



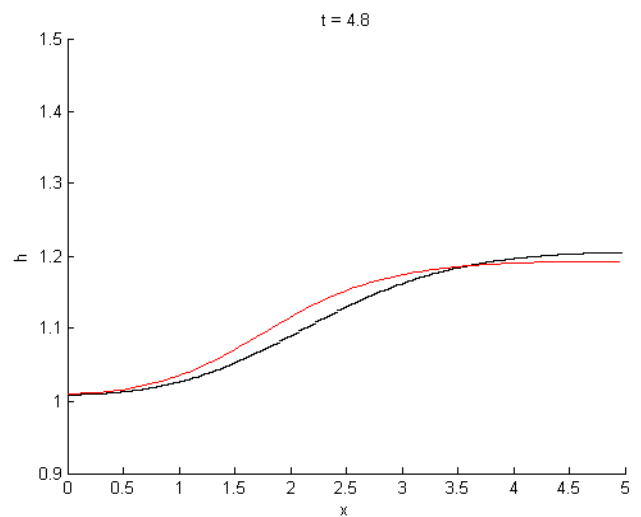
$\Delta x = 0.10$ m, $\Delta t = 0.01$ s



$\Delta x = 0.025$ m, $\Delta t = 0.01$ s



$\Delta x = 0.10$ m, $\Delta t = 0.04$ s



$\Delta x = 0.025$ m, $\Delta t = 0.04$ s