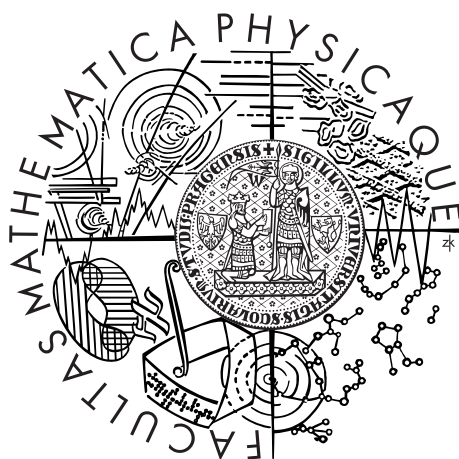


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Jan Kotek

### Numerické řešení fyzikálních úloh vedoucích k soustavě parciálních diferenciálních rovnic metodou konečných prvků – úvod do problematiky

Astronomický ústav UK

Vedoucí bakalářské práce: Mgr. Miroslav Bárta, Ph.D.

Studijní program: Fyzika

Studijní obor: Obecná fyzika

Praha 2014

Chtěl bych tímto poděkovat Miroslavu Bártovi za velmi zajímavé téma a laskavé vedení. Děkuji i všem těm, kteří mě k astronomii a fyzice přivedli či v uplynulých letech ve studiu podporovali.

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

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

V Praze dne 13. května 2014

Podpis autora

Název práce: Numerické řešení fyzikálních úloh vedoucích k soustavě parciálních diferenciálních rovnic metodou konečných prvků – úvod do problematiky

Autor: Jan Kotek

Katedra: Astronomický ústav UK

Vedoucí bakalářské práce: Mgr. Miroslav Bárta, Ph.D., Astronomický ústav AV ČR, v.v.i.

Abstrakt: V práci je nastíněna problematika parciálních diferenciálních rovnic, jejich klasifikace a jsou představeny numerické metody řešení těchto rovnic: metoda konečných diferencí, metoda konečných objemů a zvláštní pozornost je věnována metodě konečných prvků (FEM). Metoda FEM je aplikována na řešení několika ilustrativních příkladů: rovnici vedení tepla, Poissonovu rovnici a rovnici šíření elektromagnetické vlny disperzním prostředím. Existuje několik formulací FEM pro PDE. V této práci byly studovány dvě nejpoužívanější, Galerkinova metoda a LSFEM. Výsledky jsou pokud možno ověřeny pomocí analytického řešení, v případě rovnice vedení tepla jednak pomocí tepelného jádra a jednak pomocí Fourierových řad. Krátce jsou zmíněny adaptivní metody, konkrétně metoda *hp*FEM a C++ knihovna pro její implementaci HERMES.

Klíčová slova: PDR - numerické řešení, FEM, LSFEM, EM vlna v plazmatu

Title: Numerical solution of physical problems leading to the system of PDEs by the Finite Element Method – an introduction

Author: Jan Kotek

Department: Astronomical institute of Charles University

Supervisor: Mgr. Miroslav Bárta, Ph.D., Astronomical Institute, Academy of Sciences of the Czech Republic

Abstract: This thesis starts with a description of partial differential equations and their classification. The main aim is to describe numerical methods for their solving: the Finite Difference Method, the Finite Volume Method via finite difference formulation and especially the Finite Element Method. The Galerkin weighted residual method is applied to the heat equation and the Least Square FEM is applied to the Poisson equation as well as to the electromagnetic pulse propagation in cold electron plasma or dispersion environment in general. Results are compared with an analytical solution if available. The heat equation is solved using both fundamental solution and separation of variables leading to the Fourier series. Adaptive methods and *hp*FEM C++ numerical library HERMES are briefly mentioned too.

Keywords: PDE - Numerical solution, FEM, LSFEM, EM wave in plasma

# Obsah

Úvod	2
<b>1 Úvod do parciálních diferenciálních rovnic</b>	<b>3</b>
1.1 Laplaceova rovnice	4
1.2 Rovnice vedení tepla	5
1.3 Vlnová rovnice	5
<b>2 Numerické metody řešení parciálních diferenciálních rovnic</b>	<b>6</b>
2.1 Metoda konečných diferencí	6
2.1.1 Metoda konečných objemů	8
2.2 Metoda konečných prvků	8
2.2.1 Rayleigh-Ritzova metoda	9
2.2.2 Metoda vážených reziduí: speciálně Galerkinova	10
2.2.3 LSFEM	11
2.2.4 <i>hp</i> FEM	11
<b>3 Aplikace</b>	<b>14</b>
3.1 Poissonova rovnice	14
3.2 Vedení tepla	15
3.3 Vlna v plazmatu	19
Závěr	24
Seznam použité literatury	25
Seznam použitých zkratk	26
Přílohy	27

# Úvod

Počítačové modelování umožnilo zkoumat jevy, jejichž výzkum by byl jinak značně složitý, výrazně také zjednodušilo a zlevnilo navrhování nových výrobků v průmyslu. Fyzikální problémy, které jsou řešeny pomocí numerické analýzy, jsou často formulovány pomocí diferenciálních rovnic. S rozvojem fyziky a techniky začalo rychle přibývat problémů, které nebylo možné řešit analyticky. Byla snaha hledat alespoň řešení přibližná. Některé numerické metody byly sice známy již od antiky a na sklonku dvacátého století byly i ty modernější už celkem běžně aplikovány například v astronomii nebo ve stavební mechanice, ale nedostatek výpočetní kapacity značně omezoval jejich rozvoj. Přesto byla v první polovině 20. století rozvinuta teorie metody konečných diferencí a byly zakládány první výzkumné týmy věnované numerické analýze. Za jednu ze základních prací konečných diferencí lze považovat článek R. Couranta, K.O Fridrichse a H. Lewyho *Über die partiellen Differenzgleichungen der mathematischen Physik*. V něm došlo k roztřídění parciálních diferenciálních rovnic a byly studovány podmínky stability numerických řešení [7]. Skutečný počátek moderní numerické analýzy potom odstartovaly válečné potřeby během druhé světové války, mimo jiné počítání drah dělostřeleckých granátů v různých meteorologických podmínkách, za kterýmž účelem byly stavěny první programovatelné počítače. V numerické analýze často převádíme diferenciální rovnice na soustavu lineárních algebraických rovnic. Velmi důležitá byla proto práce H. Goldstina a von Neumanna *Numerical inverting of matrices of high order*, Alana Turinga a dalších v oblasti numerické lineární algebry. [7]

R. Courant stál i u zrodu metody konečných prvků díky článku *Variational methods for the solution of problem of equilibrium and vibrations*, přesto se tato metoda založená na matematice rozvinuté na přelomu 19. a 20. století (Rayleigh, Ritz, Galerkin) začala plně rozvíjet až od konce 50. let. Už Courant si všiml obrovské výhody metody konečných prvků - volnosti diskretizace [7].

Právě variabilita konečných prvků může za to, že se staly první volbou při počítání deformací ve strojírenství, jeden z prvních článků publikovaný v *Aeronautical Science Journal* se zabýval analýzou namáhání materiálů v letadlech. Později se ruku v ruce se zlepšováním teorie a zrychlováním algoritmů začaly konečné elementy uplatňovat i v mechanice tekutin a v dalších oblastech. Konečně ke konci 20. století I.Babuska a další rozvinuli adaptivní metodu *hpFEM* schopnou velmi rychlé (až exponenciální) konvergence [1].

Byla snaha práci pojmut tak, aby byla vhodným prostředkem pro rychlé základní zorientování v problematice. Měla by čtenáři usnadnit uchopení složitějších problémů v konkrétních aplikacích (např. MHD modelování v astrofyzikálním plazmatu). Podle toho byly sestaveny první dvě kapitoly, kde jsou primárně zařazeny poznatky používané později při řešení konkrétních rovnic ale jsou zmíněny i související a pokročilejší metody. Hlavním výsledkem práce je potom představení řešení PDR pomocí neadaptivních FEM v kapitole 2.2 společně s konkrétními ilustračními aplikacemi ve 3. kapitole.

# 1. Úvod do parciálních diferenciálních rovnic

*Parciální diferenciální rovnice* (PDR) je diferenciální rovnice, v níž se vyskytují derivace podle více proměnných. Přesněji lze PDR definovat:

**Definice** Necht'  $k \geq 1, k \in \mathbb{N}$  a necht'  $\Omega \in \mathbb{R}^n$  je otevřená množina. Potom

$$F(D^k u(x), D^{k-1} u(x), \dots, Du(x), u(x), x) = 0 \quad (x \in \Omega) \quad (1.1)$$

se nazve parciální diferenciální rovnicí  $k$ -tého řádu,

$$F : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \dots \times \mathbb{R}^n \times \mathbb{R} \times \Omega \rightarrow \mathbb{R}$$

je dáno,  $D^k$  značí množinu parciálních derivací řádu  $k$ , a  $u : \Omega \rightarrow \mathbb{R}$  je neznámá (řešení rovnice 1.1). [2]

Pokud je navíc rovnice ve tvaru  $\sum_{|\alpha|=k} a_\alpha(x) D^\alpha u = f(x)$  kde  $\alpha$  značí multiindex, hovoříme o *lineární* PDR. Rovnici s  $f(x) = 0$  nazveme *homogenní*.

Na  $u$  klademe okrajové podmínky, to jest požadavek na hodnotu  $u$  (*Dirichletova*) nebo hodnotu derivací  $u$  (*von Neumannova*) případně jejich kombinaci na hranici  $\partial\Omega$ . Pokud je to vůbec možné, tak pro jednoznačnost řešení rovnice  $2m$ -tého řádu požadujeme Neumannovy okrajové podmínky řádu  $m, \dots, 2m - 1$  nebo Dirichletovy řádu  $0, \dots, m - 1$ . Klasickým řešením rozumíme řešení, jehož všechny derivace vystupující v konkrétní formulaci PDR jsou spojité. Protože nalézt takové řešení je často obtížné, může být vhodné tuto podmínku oslabit. Mluvíme potom o *slabém řešení* PDR.

Často se zajímáme o úlohy typu

$$\mathcal{D}[u] = f \quad (1.2)$$

kde  $\mathcal{D}$  je diferenciální operátor s konstantními koeficienty, případně soustavy takovýchto rovnic. *Fundamentálním řešením* úlohy 1.2 nazýváme funkci  $U$  takovou, pro kterou platí

$$\mathcal{D}[U] = \delta_0 \quad (1.3)$$

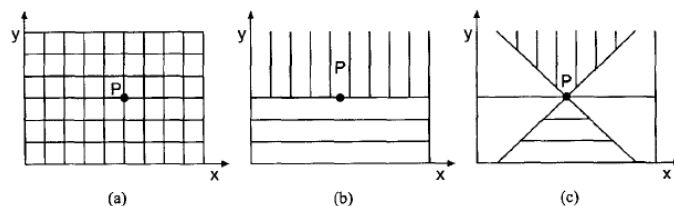
kde  $\delta_0$  je Diracova distribuce. Pro obecnou pravou stranu lze získat řešení 1.2 díky vztahu

$$u = U * f \quad (1.4)$$

pokud má tato konvoluce smysl.

Ve fyzikálních úlohách běžně řešíme lineární PDR s konstantními koeficienty maximálně druhého řádu.

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial x} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + Fu + G = 0 \quad (1.5)$$



Obrázek 1.1: Ilustrace rozdělení oblasti  $\Omega$  pro potřeby klasifikace PDE pro a) eliptické b)parabolické a c)hyperbolické rovnice [3]

Ve shodě s geometrickou analogií dělíme tyto PDR na základě znaménka výrazu  $\kappa = B^2 - 4AC$  na *eliptické* ( $\kappa < 0$ ), *parabolické* ( $\kappa = 0$ ) a *hyperbolické* ( $\kappa > 0$ ). Podstata dělení spočívá v tom, na jaké části  $\Omega$  závisí hodnota nějakého bodu v ní, resp. jakou část této množiny ovlivní změna hodnoty v daném bodě. Hranici této oblasti lze vyjádřit jako přímku splňující rovnici

$$\frac{dy}{dx} = \frac{-B \pm \kappa^{1/2}}{2A} \quad (1.6)$$

Pro případ eliptické rovnice nemá žádné řešení, porucha v daném bodě A ovlivní všechny body v  $\Omega$  a naopak. Z toho vyplývá, že tyto rovnice nebudou popisovat časový vývoj fyzikálního systému, ale spíše systémy v rovnováze. Pro rovnice parabolické má rovnice právě jedno řešení, prostor  $\Omega$  je tak vzhledem k hodnotě v A rozdělen na dvě poloviny, kdy na jedné polovině závisí hodnota v A a naopak hodnoty v druhé polovině závisí na hodnotě v bodě A. To znamená, že rovnice může popisovat časový vývoj systému, nicméně rychlost šíření informace je v takovém případě nekonečná. Dalo by se říci, že je to extrémní případ hyperbolické rovnice. V případě hyperbolické rovnice existují navíc oblasti které nemají na hodnotu v A žádný vliv ani jím nejsou ovlivňovány. Tomu odpovídá konečná rychlost šíření informace. 1.1 Tuto klasifikaci lze v souladu s předchozím zobecnit i pro rovnice jiného řádu, případně pro soustavy PDR.[3]

Ve zbylé části práce se budeme zabývat zejména numerickým řešením takovýchto rovnic a soustav. Způsob jakým k řešení dospějeme, bude do značné míry záviset právě na typu PDR. V následujících oddílech budou proto tyto tři typy PDR ilustrovány na známých konkrétních příkladech.

## 1.1 Laplaceova rovnice

Příkladem eliptické PDR je Laplaceova rovnice

$$\Delta u(x) = 0 \quad (1.7)$$

kde  $x \in \Omega$  a  $\Delta$  je Laplaceův operátor. Rovnice s nenulovou pravou stranou se nazývá Poissonova. Jejím typickým použitím je vyšetřování fyzikálního potenciálu  $\varphi$  v prostředí s rozložením náboje daným pravou stranou  $\frac{\rho}{\epsilon}(\vec{x})$ .

V 1D ( $n = 1$ ) fundamentální řešení rovnice 1.7 snadno nahlédneme, pro vyšší dimenze [2]:

$$U(x) = \begin{cases} -\frac{|x|}{2}, & n = 1 \\ -\frac{\ln|x|}{2\pi}, & n = 2 \\ -\frac{|x|^{2-n}}{\beta(n-2)}, & n > 2 \end{cases} \quad (1.8)$$



kde  $\beta(n)$  značí povrch  $n$  rozměrné jednotkové sféry.

## 1.2 Rovnice vedení tepla

Rovnice vedení tepla je typickou ukázkou parabolické rovnice.

$$-\alpha \Delta u + \frac{\partial u}{\partial t} = 0 \quad (1.9)$$

Koeficient  $\alpha$  značí v případě, že je rovnice používána k popisu vedení tepla tepelnou difuzivitou, tj.  $\alpha = \frac{\lambda}{c_p \rho}$  kde  $\lambda$  značí tepelnou vodivost,  $c_p$  měrnou tepelnou kapacitu a  $\rho$  hustotu. Při řešení časově závislých PDR obvykle volíme počáteční podmínku (t.j. okrajovou podmínku v části hranice  $\partial\Omega$  kde  $t = 0$ ) jako Dirichletovu, což si lze představit jako rozložení teploty  $u$  v tělese. Pro zbylou část hranice můžeme volit například opět Dirichletovu okrajovou podmínku (přiložení tělesa k tepelnému rezervoáru) nebo třeba Neumannovu, pomocí které lze nastavit tepelný tok na okraji. Dokonale izolovanému tělesu odpovídá  $\partial_x u = 0$ . Fundamentální řešení  $n$  dimenzionální rovnice vedení tepla s  $\alpha = 1$ :

$$U(x, t) = \frac{H(t)}{(4\pi t)^{n/2} e^{-\frac{|x|^2}{4t}}} \quad (1.10)$$

kde  $H(t)$  je Heavisideova funkce. Pro danou počáteční podmínku platí

$$u(x, t) = \int_{\mathbb{R}^n} U(x - y, t) u_0(y) dy \quad (1.11)$$

K analytickému řešení RVT s homogenními okrajovými podmínkami lze použít Fourierovy metody separace proměnných. Vyšetřujeme 1D homogenní RVT s homogenními okrajovými podmínkami. Vyjádříme-li počáteční podmínku pomocí Fourierovy řady

$$u(x, 0) = \sum_{n=1}^{\infty} a_n \sin \frac{n\pi x}{l} \quad (1.12)$$

Potom platí [4]:

$$u(x, t) = \sum_{n=1}^{\infty} a_n \sin \frac{n\pi x}{l} e^{-\frac{n^2 \pi^2 \alpha t}{l^2}} \quad (1.13)$$

## 1.3 Vlnová rovnice

Mezi hyperbolické rovnice se řadí vlnová rovnice.

$$-\Delta u + \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = 0 \quad (1.14)$$

kde  $c$  značí rychlost šíření vlny. Okrajové podmínky mají v případě vlnové rovnice aplikované na strunu význam pevného (Dirichletova) a volného konce ( $\Delta u = 0$ ).

Rovnici lze faktorizovat jako soustavu dvou PDR prvního řádu

$$\frac{\partial u}{\partial t} - \frac{\partial v}{\partial x} = 0 \quad (1.15)$$

$$\frac{\partial v}{\partial t} - \frac{\partial u}{\partial x} = 0 \quad (1.16)$$

## 2. Numerické metody řešení parciálních diferenciálních rovnic

Nalézt řešení PDR 1.1 nebo dokonce jejich soustavy je netriviální i pro jednoduché rovnice zmíněné v minulé kapitole. Proto je často vhodné řešit PDR numericky. To znamená nalézt přibližné řešení PDR v diskrétních bodech (nebo na elementech) takové, že v limitě nekonečně husté diskretizační sítě splňuje danou PDR. Ve zbytku kapitoly budou představeny tři metody numerického řešení, metoda konečných diferencí, objemů a metoda konečných prvků.

### 2.1 Metoda konečných diferencí

Metoda konečných diferencí vychází z definice derivace.

$$\frac{\partial u(x)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{u(x - \Delta x) - u(x)}{\Delta x} \quad (2.1)$$

Nejprve provedeme *diskretizaci* oblasti  $\Omega$ , tu rozdělíme do ortogonální mříže. Místa, kde se jednotlivé mříže protínají, nazveme uzly. Ve smyslu rovnice 2.1 ztotožníme  $\Delta x$  s délkou mříže mezi dvěma uzly. Vidíme, že pro  $\Delta x \rightarrow 0$  se stává oblast  $\Omega$  spojitou. Pokud danou PDR zapíšeme ve tvaru pro řešení pomocí konečných diferencí požadujeme její *konzistenci*, tedy aby v této limitě přecházela v PDR z níž vychází.

Pokud je hodnota  $\Delta x$  nenulová, dochází při výpočtu derivace k zanedbání změn mezi jednotlivými body a ke vzniku chyby. Pokud je tato chyba omezená, říkáme, že schéma použité k výpočtu je *stabilní*.

Derivace lze počítat pomocí různě přesných schémat. Taylorovým rozvojem v  $u_{i\pm 1}$  [1]

$$u_{i\pm 1} = u_i + \sum_{j \geq 1} \frac{(\pm \Delta x)^j}{j!} \frac{\partial^j u(x_i)}{\partial x^j} \quad (2.2)$$

dostáváme

$$\frac{\partial u(x)}{\partial x} = \frac{u(x_{i+1}) - u(x_i)}{\Delta x} + O(\Delta x) \quad (2.3)$$

$$\frac{\partial u(x)}{\partial x} = \frac{u(x_{i+1}) - u(x_{i-1}))}{2\Delta x} + O(\Delta x^2) \quad (2.4)$$

Z Taylorova rozvoje můžeme snadno odvodit i schémata pro vyšší derivace a s menší chybou. To proto, abychom dosáhli *konvergence*, tedy aby pro  $\Delta x \rightarrow 0$  numerické řešení PDR přecházelo v analytické. Schémata dělíme na explicitní a implicitní podle toho, jestli k počítání derivací používáme už nové (implicitní) anebo pouze staré hodnoty veličin (explicitní). V případě implicitních metod řešíme všechny uzly najednou jako soustavu. Jedno z používaných stabilních implicitních schémat je Crank-Nicholsonovo. Pro ilustraci uvedeme použití na rovnici  $u_t - u_x = 0$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2} \left( \frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} + \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (2.5)$$

kde horní index značí čas.

K určení stability konkrétního schématu se používá von Neumannovy analýzy stability, která spočívá v rozvoji chyb do Fourierových řad. Budeme postupovat podle [1]. Necht  $\overline{\mathbf{U}}^n$  a  $\mathbf{U}^n$  značí vektor přesných respektive spočtených řešení ve všech uzlech v čase  $n$  a chyba  $\epsilon$  je jejich rozdíl. Můžeme psát

$$\epsilon^{n+1} = \mathbf{C}\epsilon^n \quad (2.6)$$

kde lineární operátor  $\mathbf{C}$  je příslušný danému schématu. Například pro jedno ze schémat řešící rovnici vedení tepla

$$u_i^{n+1} = u_i^n + \underbrace{\frac{\alpha\Delta t}{\Delta x^2}}_{\zeta} (u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (2.7)$$

dostaneme  $\epsilon$  s na vedlejších digonálách a  $1 - 2\zeta$  na hlavní. Chybu můžeme stejně jako řešení rozložit do Fourierovy řady

$$\epsilon_i^n = \sum_{j=-N}^N \bar{\epsilon}_j^n e^{\frac{Ij\pi i}{N}} \quad (2.8)$$

kde pro přehlednost značení  $I = \sqrt{-1}$  a  $N = \frac{l}{2\Delta x}$ ,  $l$  je délka množiny na níž problém řešíme. Dosadíme-li toto vyjádření do schématu 2.7 a vyjádříme-li  $\phi = \frac{j\pi}{N}$  dosaneme

$$\bar{\epsilon}^{n+1} = \bar{\epsilon}^n (1 + \zeta(e^{I\phi} - 2 + e^{-I\phi})) \quad (2.9)$$

což už pouze dosadíme do podmínky stability

$$\left| \frac{\bar{\epsilon}^{n+1}}{\bar{\epsilon}^n} \right| \leq 1 \quad (2.10)$$

a upravíme na

$$\zeta \leq \frac{1}{1 - \cos \phi} \quad (2.11)$$

Tak dostáváme dostáváme podmínku

$$0 \leq \zeta \leq \frac{1}{2} \quad (2.12)$$

Dirichletovu okrajovou podmínku lze v případě konečných diferencí aplikovat snadno nastavením konkrétní hodnoty okrajového uzlu. Jedna z možností pro aplikaci Neumannovy okr. podmínky je přidat uzly navíc. Vezměme si například podmínku

$$\frac{du}{dx}(x = \max) = 1 \quad (2.13)$$

dostaneme vztah

$$\frac{u_{i_{\max}+1} - u_{i_{\max}-1}}{2\Delta x} = 1 \quad (2.14)$$

který dosadíme do vyjádření dané diferenciální rovnice. Tím získáme okrajovou podmínku ve tvaru vhodném pro numerický výpočet. [1]

Konečné diference budou dále hrát roli u metody konečných prvků při diskretizaci v časové oblasti.

### 2.1.1 Metoda konečných objemů

K metodě konečných objemů lze přistupovat dvěma způsoby. Buď ji lze formulovat pomocí metody konečných diferencí nebo pomocí metody konečných prvků, kterou budeme diskutovat později. Zde pouze krátce představíme formulaci pomocí konečných diferencí. Základem je množina diskretizovaná stejně jako v případě normální FDM. Nejprve řešenou rovnici zintegrujeme. Doprostřed mezi jednotlivé uzly přidáme myšlené kontrolní plochy  $\partial V$ , které využijeme pro potřeby přenosu toku tj.  $\partial_x u$ . Výsledná rovnice (po integraci) v diskretizovaném tvaru bude rovnicí zachování. Například jednoduchou rovnicí

$$\frac{\partial^2 u}{\partial x^2} = 1 \quad (2.15)$$

po zintegrování přepíšeme jako rovnici kontinuity [1]

$$\sum_{\partial V} \frac{\Delta u}{\Delta x} = \Delta x \quad (2.16)$$

kteřou ještě rozepíšeme do tvaru vhodného pro výpočet

$$\frac{(u_{i+1} - u_i) - (u_i - u_{i-1})}{\Delta x} = \Delta x \quad (2.17)$$

Dirichletovy okrajové podmínky se aplikují stejně jako v případě FDM, při aplikaci Neumannovy okrajové podmínky prostě jen připočteme před finální úpravou ke krajnímu uzlu požadovaný tok. Metoda konečných objemů je přirozeně formulovaná pro (soustavy) PDR, které lze interpretovat jako zákony zachování v konzervativním tvaru. Další výhodou metody konečných objemů se projeví při řešení více dimenzionálních problémů a spočívá v někdy menší závislosti na poloze jednotlivých uzlů a tedy i na zvolené diskretizaci [1].

## 2.2 Metoda konečných prvků

Charakteristickou vlastností metody konečných prvků (Finite Element Method - FEM) je větší variabilita diskretizace. To se s výhodou projeví všude tam, kde je diskretizace do pravoúhlé a pravidelné mříže nevyhovující. Při FEM se oblast diskretizuje na množství prvků, což jsou n-dimenzionální útvary s plochými stěnami pokrývající celou vyšetřovanou oblast. Vrcholy se nazývají uzly. Typickým příkladem může být rozčlenění roviny do trojúhelníkové sítě. Jednotlivé uzly, k nimž se stejně jako u metody konečných diferencí vztahují hodnoty proměnné, lze indexovat dvěma způsoby - *globálním indexem* který je unikátní pro každý uzel a *lokálním indexem* kterým jsou očíslovány všechny uzly příslušící danému prvku. Mezi oběma způsoby indexování lze snadno přecházet pomocí binární matice  $\Delta_{N\alpha}^e$ ,  $N$  tý uzel elementu  $e$  zapsaný lokálně jako  $z_N^e$  lze pomocí ní vyjádřit jako

$$z_N^e = \Delta_{N\alpha}^e Z_\alpha \quad (2.18)$$

kde využíváme Einsteinovy sumační konvence (tu budeme používat i nadále) a  $Z_\alpha$  je uzel vyjádřený globálně.

K hledání řešení PDR přistupovat buď variačně a nebo pomocí metody (vážených) reziduí. Obě metody byly původně formulovány k hledání analytického řešení PDR. Protože je to často nemožné, my se nebudeme pokoušet najít řešení, které splní danou PDR všude, ale spíše větší množství řešení, která dohromady pokryjí celou množinu na níž PDR řešíme a která jí splní alespoň přibližně v daném prvku. Řešení se pokusíme hledat ve tvaru

$$u(x) = \bigcup_e \Phi_N^e(x) u_N^e = \Phi_\alpha u_\alpha \quad (2.19)$$

kde  $\Phi(x)$  je *bázová funkce*. V této práci budeme za bázovou funkci volit především lineární funkce

$$\Phi(x)_1^e = \frac{x}{h} \quad \Phi(x)_2^e = 1 - \frac{x}{h} \quad (2.20)$$

kde  $h$  je délka jednoho prvku. Nicméně, jako bázové funkce mohou být voleny i polynomy vyšších řádů, případně jiné funkce (např. goniometrické).

Nakonec si ukážeme podmínku stability pro metodu konečných prvků [1]. Metoda konečných prvků vede na soustavu rovnic ve tvaru  $\mathbf{K} \cdot \mathbf{u} = \mathbf{F}$ . V Banachově prostoru  $L_p$  definujeme  $p$ -normu

$$\|e\|_{L_p} = \left( \int e^p dx \right)^{\frac{1}{p}} \quad (2.21)$$

Pro diskrétní hodnoty potom namísto integrálu použijeme sumu.  $L_2$  norma potom v podstatě přechází v „euklidovskou“ normu. Takovou normu budeme značit  $\| \cdot \|$ . Zde si zdefinujeme maticovou normu  $n \times n$  matice  $\mathbf{M}$  jako

$$\|\mathbf{M}\| = \min \left\{ c : c \in \mathbb{R}, c \geq \frac{\|\mathbf{M}\mathbf{u}\|}{\|\mathbf{u}\|} \right\} \quad (2.22)$$

kde  $\mathbf{u}$  je libovolný vektor s  $n$  prvky a s nenulovou normou. Nakonec zdefinujeme číslo

$$\Upsilon(\mathbf{K}) = \|\mathbf{K}\| \|\mathbf{K}^{-1}\| \quad (2.23)$$

Řekneme, že soustava  $\mathbf{K} \cdot \mathbf{u} = \mathbf{F}$  je dobře podmíněná, pokud  $\Upsilon$  je malé.

## 2.2.1 Rayleigh-Ritzova metoda

Existuje-li nějaký způsob, jak daný fyzikální zákon resp. rovnici formulovat variačně v exaktní formě (to neplatí obecně například v hydrodynamice), lze využít variační metody podobně jako v mechanice využíváme principu minimalizace virtuální práce. Jako příklad lze uvést řešení rovnice

$$u''(x) - 2 = 0 \quad (2.24)$$

na intervalu  $(0,1)$ . Po vynásobení  $u'$  rovnici přepíšeme do tvaru

$$0 = \left( \frac{(u')^2}{2} - 2u \right)' \quad (2.25)$$

rovnici vyintegrujeme podle  $x$  a dostaneme funkcionál:

$$I(x, u(x), u'(x)) = \int_0^1 \left[ \frac{(u')^2}{2} - 2u \right] dx \quad (2.26)$$

Snadno ověříme, že po dosazení do Euler-Lagrangeovy rovnice dostaneme původní rovnici 2.24. Pokud aproximujeme řešení  $u(x)$  parametry  $a, b, \dots$  budeme minimalizovat integrál 2.26 vzhledem k těmto parametrům [3]

$$\delta I(x, u(x, a, b, \dots), u'(x, a, b, \dots)) = \frac{\partial I}{\partial a} \delta a + \frac{\partial I}{\partial b} \delta b + \dots \quad (2.27)$$

což je splněno jen pro  $\frac{\partial I}{\partial a} = \frac{\partial I}{\partial b} = 0$  Při numerickém řešení Rayleigh-Ritzovou metodou se zajímáme jen o řešení uvnitř jednoho prvku a používáme za parametry hodnoty hledané funkce v příslušných uzlech. Pro jednodimenzionální problém proto ve smyslu značení z počátku podkapitoly

$$\delta I = \frac{\partial I}{\partial u_\alpha} \delta u_\alpha = 0 \quad (2.28)$$

kde  $\delta u_\alpha$  je libovolné, tedy nulové musejí být i jednotlivé derivace. Dosazením 2.19 do 2.28 dostaneme soustavu rovnic, kterou vyřešíme.

## 2.2.2 Metoda vážených reziduí: speciálně Galerkinova

Pokud nelze přímo využít variačního principu, můžeme definovat *reziduum*  $R$  jako člen, který je třeba přičíst k PDR po dosazení přibližného řešení aby platila rovnost. Dále uvažujeme váhové funkce, které jsou v případě Galerkinovy metody v 1D identické s funkcemi definovanými v 2.20. Princip Galerkinovy metody vážených reziduí spočívá v minimalizaci skalárního součinu bázevých funkcí a rezidua k nule [1]:

$$\int_{\Omega} \Phi_w R d\Omega = 0 \quad (2.29)$$

Zkusme nyní pro jednoduchost aplikovat tuto metodu na problém 2.24. Po dosazení do 2.29 a integraci per-partes dostaneme:

$$\left[ \Phi_w \frac{du}{dx} \right]_0^1 - \int_0^1 \frac{d\Phi_w}{dx} \frac{du}{dx} dx - \int_0^1 2\Phi_w dx = 0 \quad (2.30)$$

kde první člen představuje fakticky von Neumannovu okrajovou podmínku. V druhém členu nahradíme  $u$  přibližným řešením  $\Phi_w u_\alpha$ . Získáme tak soustavu rovnic kterou lze zjednodušeně zapsat  $K_{\beta\alpha} u_\alpha = F_\beta + G_\beta$ . Matici  $K$  nazveme *matice tuhosti* a vektor  $F$  *vektor zatížení*. Pokud bychom chtěli namísto von Neumannovy okrajové podmínky předepsat Dirichletovu, potom ji pouze zapíšeme na místo první a poslední rovnice soustavy. Vyřešením soustavy dostaneme numerické řešení zadané diferenciální rovnice.

### 2.2.3 LSFEM

Integrace po částech která byla dosud nezbytná, může být poměrně komplikovaná a v případě složitější soustavy i dosti nepraktická. Můžeme se jí vyhnout použitím metody LSFEM (Least Square FEM). Důvod jejího pojmenování *Nejmenších čtverců* (Least Square FEM) bude zřejmý z následujícího. Vycházíme z toho, že platí

$$Lu = S \quad (2.31)$$

kde  $L$  je nějaký lineární diferenciální operátor a  $S$  představuje pravou stranu. Z toho vyplývá, že pro  $u$  řešící 2.31 musí platit

$$I(u) = \int_{\Omega} (Lu - S)^2 dx = 0 \quad (2.32)$$

V případě numerického řešení se budeme snažit funkcionál  $I(u)$  („čtverec“ reziduí) minimalizovat.  $u$  nahradíme

$$u(x) = \overline{u(x)} + tv(x) \quad (2.33)$$

V souladu s nutnou podmínkou minima funkcionálu 2.32 budeme požadovat pro každé  $v(x)$

$$I(u) = \int_{\Omega} (L\bar{u}(x) - S(x)) \cdot Lv(x) dx = 0 \quad (2.34)$$

Za  $v(x)$  zvolíme bázové funkce a  $\bar{u}(x)$  jimi budeme aproximovat. Získáme tak pro každý element soustavu dvou rovnic

$$u_0^e \int_e \mathbf{L}\Phi_0^e(x) \cdot \mathbf{L}\Phi_0^e(x) dx + u_1^e \int_e \mathbf{L}\Phi_0^e(x) \cdot \mathbf{L}\Phi_1^e(x) dx = \int_e \mathbf{L}\Phi_0^e(x) \cdot \mathbf{S}(x) dx \quad (2.35)$$

$$u_0^e \int_e \mathbf{L}\Phi_1^e(x) \cdot \mathbf{L}\Phi_0^e(x) dx + u_1^e \int_e \mathbf{L}\Phi_1^e(x) \cdot \mathbf{L}\Phi_1^e(x) dx = \int_e \mathbf{L}\Phi_1^e(x) \cdot \mathbf{S}(x) dx \quad (2.36)$$

Sečtením rovnic pro společné vnitřní uzly ( $u_1^e$  označuje stejný uzel jako  $u_0^{e+1}$ ) dostaneme pro 1D útvar o  $N$  elementech soustavu  $N + 1$  rovnic, kterou už zbývá pouze vyřešit.

### 2.2.4 hpFEM

Zatímco v této práci aproximujeme jen lineárními funkcemi a uzly máme rovnoměrně rozmístěné, v některých případech - zejména u multiškálových problémů - tento postup k žádoucím výsledkům nevede. Příkladem může být studium turbulence nebo magnetické rekonexe v rozlehlých systémech. V takových případech je vhodné používat adaptivní metody. *hpFEM* je adaptivní metoda konečných prvků kombinující  $h$  zjemňování prvků a  $p$  zvyšování řádu ortogonálních polynomů používaných k aproximaci (v neadaptivním případě se takové metody používají např. Čebyševovy, Legendreovy a jiné polynomy nazývají *spektrální*). Při zjemňování prvků definujeme maximální přípustnou chybu řešení, kterou regulujeme právě zjemňováním sítě v daném místě. Pro šetření výpočetního času se používá i opačný proces, aby síť nebyla zbytečně hustá. [1]

Při použití  $hp$ FEM obvykle postupujeme tak, že po zvolení maximální přípustné chyby se pokusíme vyřešit problém s určitými parametry a následně spočítáme chybu. Pokud je chyba příliš velká, upravujeme  $h$  a  $p$  dokud nedosáhneme požadované přesnosti. Obtížné na  $hp$ FEM je to, že existuje mnoho možností jakými lze úpravy provést. Například C++ knihovna HERMES [6] používá k rozhodování mimo jiné informaci o přesnosti z jedné jemnější a jedné hrubší sítě. V  $d$  dimenzích pro délku prvku  $h$  při použití metody  $h$ FEM platí

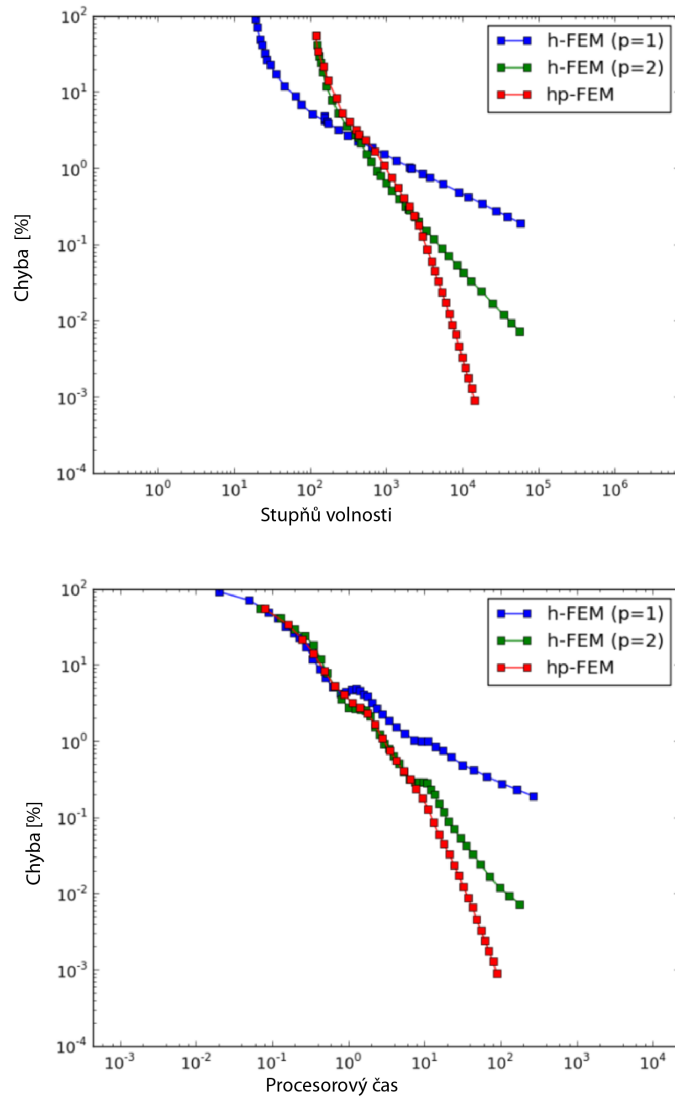
$$h = O(N^{-p/d}) \quad (2.37)$$

kde  $N$  je počet stupňů volnosti a  $p$  je stupeň polynomu (v případě rovnoměrně se zmenšujícího  $h$  je to  $O(h^p)$ ). S tím souvisí i odhad

$$\text{chyba} \approx O(N^{-p/d}) \quad (2.38)$$

Ukazuje se (obr. 2.1), že konvergence  $hp$ FEM může být v tomto smyslu až exponenciální. [6]





Obrázek 2.1: Porovnání konvergence různých adaptivních metod v závislosti na počtu stupňů volnosti a procesorového času. [6]

## 3. Aplikace

V následující kapitole je aplikována teorie rozvinutá v kapitolách předchozích. Konkrétně je na řešení Poissonovy rovnice, rovnice vedení tepla a na vlně v plazmatu ilustrována metoda konečných prvků. Pokud to bylo možné, pokusil jsem se srovnat řešení numerické s řešením analytickým. Při psaní programů jsem k manipulaci s maticemi používal C++ knihovnu Armadillo verze 3.2.3 [5].

### 3.1 Poissonova rovnice

Na příkladu poměrně jednoduché 1D Poissonovy rovnice s konstantní pravou stranou se pokusím předvést metodu LSFEM, vliv okrajových podmínek a zároveň princip, jakým lze přistupovat k řešení soustav rovnic. Uvažujme rovnici

$$\frac{\partial^2 u}{\partial x^2} = 1 \quad (3.1)$$

což může představovat například potenciál v poli s nábojovou hustotou  $-\epsilon$ . Převědeme jí na soustavu rovnic prvního řádu

$$\underbrace{\begin{pmatrix} 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial x} & -1 \end{pmatrix}}_{\mathbf{L}} \cdot \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\mathbf{U}} = \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\mathbf{S}} \quad (3.2)$$

$\mathbf{U}$  aproximujeme na elementech pomocí bázových funkcí 2.20

$$\mathbf{U}^e = \Phi_0 \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \Phi_1 \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} \quad (3.3)$$

Budeme se v principu snažit minimalizovat integrál 2.32. Protože ale  $\mathbf{U}$  je vektorová veličina, zavedeme vektory  $\mathbf{V}_k$

$$\{\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_4\} = \left\{ \Phi_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \Phi_0 \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \Phi_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \Phi_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \quad (3.4)$$

Namísto dvou rovnic 2.35 a 2.36 tak dostaneme rovnice čtyři a celkem budeme muset vyřešit soustavu  $2N - 1$  lineárních rovnic

$$\int_e \mathbf{L} \Phi_0 \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \cdot \mathbf{L} \mathbf{V}_k dx + \int_e \mathbf{L} \Phi_1 \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} \cdot \mathbf{L} \mathbf{V}_k dx = \int_e \mathbf{S} \cdot \mathbf{L} \mathbf{V}_k dx \quad (3.5)$$

což je to samé jako

$$\int_e \mathbf{L}^T \Phi_0 \cdot \mathbf{L} \mathbf{V}_k dx \cdot \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \int_e \mathbf{L}^T \Phi_1 \cdot \mathbf{L} \mathbf{V}_k dx \cdot \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \int_e \mathbf{S} \cdot \mathbf{L} \mathbf{V}_k dx \quad (3.6)$$

V soustavě lze vysledovat jistou pravidelnost, označíme-li

$$\mathbf{F}_i^e = \int_e \mathbf{L}^T \Phi_i \cdot \mathbf{S} dx \quad (3.7)$$

lze díky zavedení matice

$$\mathbf{M}_{i,j} = \int_e \mathbf{L}^T \Phi_i \cdot \mathbf{L} \Phi_j dx \quad (3.8)$$

přepsat soustavu jako

$$\begin{pmatrix} \mathbf{M}_{0,0} & \mathbf{M}_{1,0} & 0 & \cdots & 0 \\ \mathbf{M}_{0,1} & (\mathbf{M}_{1,1} + \mathbf{M}_{0,0}) & \mathbf{M}_{1,0} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \mathbf{M}_{0,1} & (\mathbf{M}_{1,1} + \mathbf{M}_{0,0}) & \mathbf{M}_{1,0} \\ 0 & \cdots & 0 & \mathbf{M}_{0,1} & \mathbf{M}_{1,1} \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ v_0 \\ \vdots \\ u_N \\ v_N \end{pmatrix} = \begin{pmatrix} \mathbf{F}_0 \\ \mathbf{F}_1 + \mathbf{F}_0 \\ \vdots \\ \mathbf{F}_1 + \mathbf{F}_0 \\ \mathbf{F}_1 \end{pmatrix} \quad (3.9)$$

Ještě zbývá vyčíslit hodnoty matic  $\mathbf{M}_{i,j}$  a vektoru  $\mathbf{F}$ . Označíme-li  $h$  délku jednoho elementu

$$\mathbf{M}_{0,0} = \begin{pmatrix} 1/h & 1/2 \\ 1/2 & 1/h + h/3 \end{pmatrix} \quad (3.10)$$

$$\mathbf{M}_{1,0} = (\mathbf{M}_{0,1})^T = \begin{pmatrix} -1/h & 1/2 \\ -1/2 & h/6 - 1/h \end{pmatrix} \quad (3.11)$$

$$\mathbf{M}_{0,0} = \begin{pmatrix} 1/h & -1/2 \\ -0.5 & 1/h + h/3 \end{pmatrix} \quad (3.12)$$

$$\mathbf{F} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.13)$$

Může se hodit pozorování, že pokud pravou stranu rovnice 3.1 modifikujeme tak, že ji vynásobíme nějakým číslem, potom stačí stačí stejným číslem vynásobit vektor  $\mathbf{F}$ . Nyní si všimněme, že soustava je blokově tridiagonální. K takové soustavě dospějeme vždy při řešení jednodimenzionálních problémů. Lze ji řešit efektivně pomocí Thomasova algoritmu 3.1 [1]. Pomocí programu 3.2 jsem pro různé okrajové podmínky a počet elementů dostal řešení zobrazená na obrázcích 3.1 a 3.2.

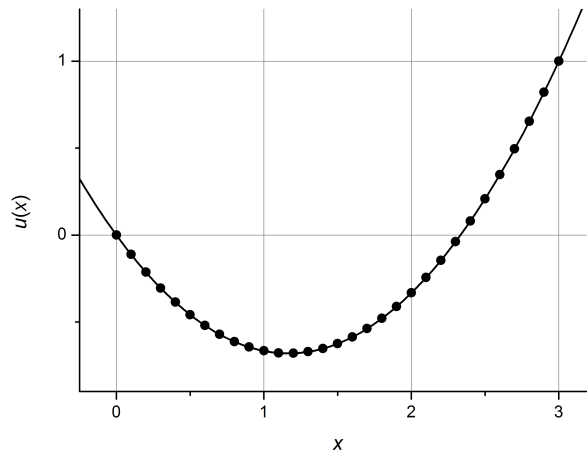
## 3.2 Vedení tepla

Rovnice vedení tepla 1.9 popisuje časový vývoj systému. Zatímco v časové oblasti použijeme konečné diference, konkrétně dopředné schéma 2.3, diskretizaci v prostorové oblasti provedeme pomocí Galerinkovy metody vážených reziduí tak, jak byla popsána v kapitole 2.2.2.

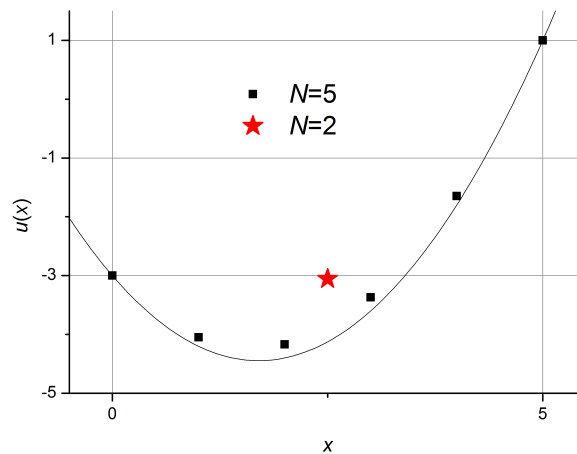
Budeme postupovat obdobně jako v [3], rovnici 2.29 zapíšeme

$$I(u(x, t)) = \int_0^l \Phi_w(\partial_t u - \partial_{xx} u) dx = 0 \quad (3.14)$$

Druhý člen lze integrovat per-partes a podobně jako v rovnici 2.30 dostaneme člen představující von Neumannovu okrajovou podmínku. Bez tohoto členu dostaneme po zpětném dosazení do 3.14 a diskretizaci pro každý prvek



Obrázek 3.1: Porovnání numerického a analytického řešení  $u(x) = -7x/6 + 3x^2$  rovnice 3.1 se zadanými okrajovými podmínkami  $u(0) = 0$  a  $u(3) = 1$  pro počet prvků  $N = 30$ .



Obrázek 3.2: Porovnání numerického a analytického řešení  $u(x) = -3 - 17x/10 + 1/2x^2$  rovnice 3.1 se zadanými okrajovými podmínkami  $u(0) = -3$  a  $u(5) = 1$  pro počet prvků  $N = 5$  a  $N = 2$ . Vidíme, že zejména v případě jednoduchých tvarů stačí jen velmi málo prvků k tomu, abychom dostali rozumné výsledky.

$$I^e(u(x, t)) = \int_e \Phi_w \partial_t u dx + \int_e \Phi_w \alpha \partial_x u dx = 0 \quad (3.15)$$

Řešení  $u(x)$  budeme aproximovat funkcemi 2.19

$$I^e(u(x, t)) = \int_e \Phi_w (\partial_t u_i \Phi_1 + \partial_t u_{i+1} \Phi_2) dx + \int_e \Phi \alpha \frac{u_{i+1} - u_i}{h} dx = 0 \quad (3.16)$$

pokud dosadíme funkce 2.19 i za  $\Phi_w$  postupně dostaneme

$$I^e(u(x, t)) = \frac{h(2u_{i-1} + \partial_t u_i)}{6} - \frac{\alpha(u_i - u_{i-1})}{h} = 0 \quad (3.17)$$

pro  $\Phi_w = \Phi_1$  a

$$I^e(u(x, t)) = \frac{h({}_t u_i + 2\partial_t u_{i+1})}{6} + \frac{\alpha(u_{i+1} - u_i)}{h} = 0 \quad (3.18)$$

pro  $\Phi_w = \Phi_2$ . Sečtením těchto dvou rovnic a dosazením vztahu 2.30 za časové derivace (pouze  $\Delta x$  nahradí  $\Delta t$ ), čímž zajistíme časový vývoj, dostaneme výsledný vztah

$$u_{i-1}^{t+\Delta t} + 4u_i^{t+\Delta t} + u_{i+1}^{t+\Delta t} = u_{i-1}^t + 4u_i^t + u_{i+1}^t + \frac{6\alpha\Delta t(u_{i-1}^t - 2u_i^t + u_{i+1}^t)}{h^2} \quad (3.19)$$

který představuje rovnici, z níž při znalosti hodnot  $u$  v čase  $t$  a (okrajových) hodnot  $u_{i-1}$  a  $u_{i+1}$  v čase  $t + \Delta t$  dostaneme hodnotu  $u_i$  v čase  $t + \Delta t$ . Při řešení rovnice vedení tepla pomocí programu 3.3 řešíme celou soustavu takových rovnic. Pro ilustraci použití fundamentálního řešení jsem zvolil počáteční podmínky

$$u(x, 0) = \begin{cases} 20 & x \in (90, 110) \\ 0 & x \in (0, 90) \cup (110, 200) \end{cases} \quad (3.20)$$

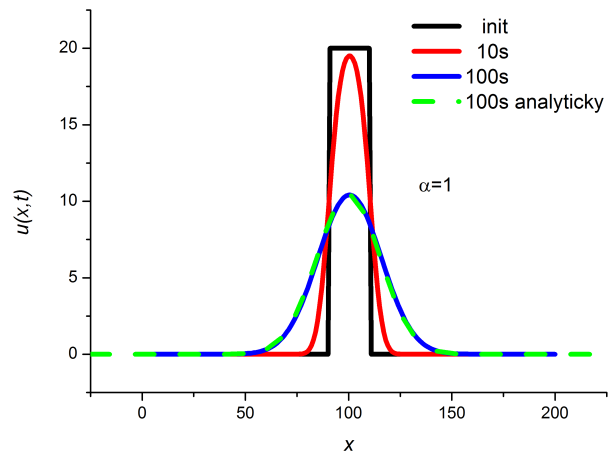
a k ilustraci metody separace proměnných

$$u(x, 0) = \begin{cases} x & x \in (0, 100) \\ 0 & x \in (100, 200) \end{cases} \quad (3.21)$$

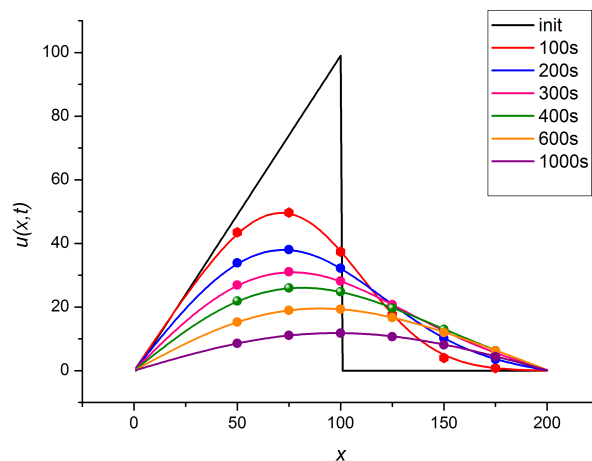
Numerické řešení jsem se pokusil ověřit pomocí řešení analytického, a to jednak pomocí Fourierových řad (obr. 3.4) a jednak pomocí znalosti fundamentálního řešení (obr. 3.3). V případě fundamentálního řešení naše veškerá práce spočívá v dosazení do rovnice 1.13. Velmi snadno (pokud jsme schopni integrál vyřešit) tak dostáváme řešení všude v prostoru pro daný čas. Oproti tomu pokud použijeme metodu separace proměnných, musíme nejprve odvodit Fourierovy koeficienty

$$a_n = -200 \frac{\pi n \cos(\frac{\pi n}{2}) - 2 \sin(\frac{\pi n}{2})}{\pi^2 n^2} \quad (3.22)$$

a následně sečíst nekonečnou řadu 1.12 abychom obdrželi výsledek pouze pro jediný bod v  $\Omega$ .



Obrázek 3.3: Porovnání numerického a analytického řešení rovnice 1.9 se zadanou počáteční podmínkou 3.17 a  $\alpha = 1$ . Protože při řešení pomocí tepelného jádra jsou meze integrálu nekonečné, zvolili jsme poměrně velké okolí v němž se nic neděje, tak aby pro naše potřeby představovalo dostatečně „nekonečno“.



Obrázek 3.4: Porovnání numerického a analytického řešení rovnice 1.9 se zadanou počáteční podmínkou 3.18 a  $\alpha = 5$ .

### 3.3 Vlna v plazmatu

V případě vlny si situaci trochu zkomplikujeme tím, že se nebude jednat o řešení obyčejné vlnové rovnice, ale pokusíme se modelovat elektro-magnetickou vlnu šířící se ve vodivém prostředí, např. v elektronovém plazmatu. Cílem je ilustrovat použití metody FEM na soustavu PDR. Při odvozování příslušné soustavy vyjdeme z Maxwellových rovnic:

$$\nabla \cdot \mathbf{B} = 0 \quad (3.23)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \quad (3.24)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (3.25)$$

kde  $\mathbf{E}, \mathbf{B}, \mathbf{j}$  značí postupně vektory intenzity elektrického pole, magnetické indukce a proudové hustoty,  $c$  potom rychlost světla ve vakuu a  $\mu_0$  permeabilitu vakua. Budeme uvažovat vlnu šířící se ve směru kartézské osy  $x$  takovou, že

$$\mathbf{E} = (0, 0, E) \quad \mathbf{B} = (0, B, 0) \quad \mathbf{J} = (0, 0, j) \quad (3.26)$$

Dosazením do rovnic 3.24 a 3.25 získáme

$$\frac{\partial B}{\partial x} = \mu_0 j + \frac{1}{c^2} \frac{\partial E}{\partial t} \quad (3.27)$$

$$\frac{\partial E}{\partial x} = \frac{\partial B}{\partial t} \quad (3.28)$$

Využijeme druhého Newtonova zákona

$$m_e n \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -ne\mathbf{E} \quad (3.29)$$

kde  $\mathbf{j} = nev$  ( $m_e$  značí hmotnost elektronu,  $e$  elementární náboj a  $n$  jeho částicovou hustotu elektronů). Z toho a z předpokladů 3.26 vyplývá

$$\frac{\partial j}{\partial t} = \frac{ne^2 E}{m_e} \quad (3.30)$$

Rovnici 3.30 výhodně přepíšeme pomocí vztahu pro plazmovou frekvenci

$$\omega_p^2 = \frac{ne^2}{m_e \epsilon_0} \quad (3.31)$$

Soustava rovnic 3.27, 3.28 a 3.30 obsahuje několik fyzikálních konstant. Provedeme renormalizaci veličin tak, aby soustava lépe vyhovovala našim budoucím potřebám, zejména aby jediná konstanta byla úměrná plazmové frekvenci. Vyjdeme z okrajové podmínky

$$E(x=0, t) = E_0 \sin(\omega_0 t) \quad (3.32)$$

Zavedeme další konstanty, představující typické škály

$$t_0 = \frac{2\pi}{\omega_0} \quad x_0 = v_0 t_0 \quad (3.33)$$

a bezrozměrné čárkované veličiny definované jako podíl skutečné hodnoty a typické škály, např.  $t' = t/t_0$ .

Dosazením do rovnice 3.28

$$\frac{E_0 \partial E'}{x_0 \partial x'} = \frac{B_0 \partial B'}{t_0 \partial t'} \quad (3.34)$$

určíme

$$B_0 = \frac{E_0}{v_0} \quad (3.35)$$

a podobně z rovnice 3.27

$$j_0 = \frac{E_0 \omega_0}{\mu_0 v_0^2 2\pi} \quad (3.36)$$

Zbývá vyjádřit rovnice 3.27, 3.28 a 3.30 pomocí čárkovaných veličin

$$\frac{\partial}{\partial t'} \begin{pmatrix} E' \\ B' \\ j' \end{pmatrix} - \frac{\partial}{\partial x'} \begin{pmatrix} B' \\ E' \\ 0 \end{pmatrix} = \begin{pmatrix} -j' \\ 0 \\ \left[ \frac{2\pi\omega_p(x,t)}{\omega_0} \right]^2 E' \end{pmatrix} \quad (3.37)$$

kde kvadrát členu v hranatých závorkách označíme  $\xi$ . Pro jednoduchost přestane veličiny čárkovat a soustavu v časové oblasti diskretizujeme pomocí Crank-Nicholsonova schématu 2.5:

$$\begin{pmatrix} 1 & -\frac{\Delta t}{2} \frac{\partial}{\partial x} & \frac{\Delta t}{2} \\ -\frac{\Delta t}{2} \frac{\partial}{\partial x} & 1 & 0 \\ -\frac{\Delta t}{2} \xi & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} E^{t+\Delta t} \\ B^{t+\Delta t} \\ j^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} E^t + \frac{\Delta t}{2} \partial_x B^t - \frac{\Delta t}{2} j^n \\ B^t + \frac{\Delta t}{2} \partial_x E^t \\ j^t + \frac{\Delta t}{2} \xi E^t \end{pmatrix} \quad (3.38)$$

Nyní v podstatě stejným způsobem jako v případě Poissonovy rovnice odvodíme vztahy pro LSFEM. Je třeba pouze počítat s tím, že nyní matice  $\mathbf{M}_{i,j}$  závisejí kvůli  $\xi(x,t)$  i na prvku a čase, jemuž přísluší. Budeme je značit  $\mathbf{M}_{i,j}^e$ . Hodnoty jednotlivých veličin potom budou mít tvar  $\mathbf{E}^N$  kde  $N$  představuje lokální index. Z rovnic 3.7 a 3.8 obdržíme:

$$\mathbf{M}_{00}^e = \begin{pmatrix} \frac{1}{12} (\xi^0)^2 h \Delta t^2 + \frac{\Delta t^2}{4h} + \frac{h}{3} & \frac{\Delta t}{2} & \frac{h \Delta t}{6} - \frac{\xi^0 h \Delta t}{6} \\ \frac{\Delta t}{2} & \frac{\Delta t^2}{4h} + \frac{h}{3} & \frac{\Delta t^2}{8} \\ \frac{h \Delta t}{6} - \frac{\xi^0 h \Delta t}{6} & \frac{\Delta t^2}{8} & \frac{h \Delta t^2}{12} + \frac{h}{3} \end{pmatrix} \quad (3.39)$$

$$\mathbf{M}_{10}^e = (\mathbf{M}_{01}^e)^T = \begin{pmatrix} \frac{1}{24} \xi^0 \xi^1 h \Delta t^2 - \frac{\Delta t^2}{4h} + \frac{h}{6} & 0 & \frac{h \Delta t}{12} - \frac{\xi^1 h \Delta t}{12} \\ 0 & \frac{h}{6} - \frac{\Delta t^2}{4h} & -\frac{\Delta t^2}{8} \\ \frac{h \Delta t}{12} - \frac{\xi^0 h \Delta t}{12} & \frac{\Delta t^2}{8} & \frac{h \Delta t^2}{24} + \frac{h}{6} \end{pmatrix} \quad (3.40)$$

$$\mathbf{M}_{11}^e = \begin{pmatrix} \frac{1}{12} (\xi^1)^2 h \Delta t^2 + \frac{t^2}{4h} + \frac{h}{3} & -\frac{t}{2} & \frac{ht}{6} - \frac{\xi^1 ht}{6} \\ -\frac{t}{2} & \frac{t^2}{4h} + \frac{h}{3} & -\frac{t^2}{8} \\ \frac{ht}{6} - \frac{\xi^1 ht}{6} & -\frac{t^2}{8} & \frac{ht^2}{12} + \frac{h}{3} \end{pmatrix} \quad (3.41)$$

Vektory  $F_i^e$  zapíšeme po složkách

$$\mathbf{F}_0^e(0) = -\frac{1}{24h} \left\{ \Delta t^2 (\xi^0 h^2 (2\xi^0 E^0 + \xi^1 E^1) + 6E^0 - 6E^1) + 2h \Delta t ((\xi^0 + 1)h(2J^0 + J^1) - 6B^1) - 4h^2 (2E^0 + E^1) \right\} \quad (3.42)$$



$$\mathbf{F}_0^e(1) = -\frac{\Delta t^2(2B^0 - 2B^1 + h(J^0 + J^1))}{8h} + \frac{1}{6}h(2B^0 + B^1) + \frac{E^1 \Delta t}{2} \quad (3.43)$$

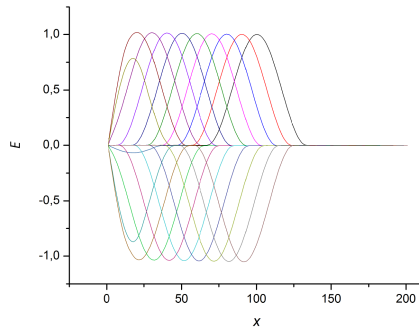
$$\begin{aligned} \mathbf{F}_0^e(2) = & \frac{1}{24h} \{2h\Delta t(2(\xi^0 + 1)E^0 + (\xi^1 + 1)E^1) + \\ & + \Delta t^2(-(3B^0 - 3B^1 + h(2J^0 + J^1))) + 4h(2J^0 + J^1)\} \end{aligned} \quad (3.44)$$

$$\begin{aligned} \mathbf{F}_1^e(0) = & \frac{1}{24h} \{\Delta t^2 (\xi^1 h^2 (\xi^0 E^0 + 2\xi^1 E^1) - 6E^0 + 6E^1) + \\ & + 2h\Delta t((\xi^1 + 1)h(J^0 + 2J^1) + 6B^0) - 4h^2(E^0 + 2E^1)\} \end{aligned} \quad (3.45)$$

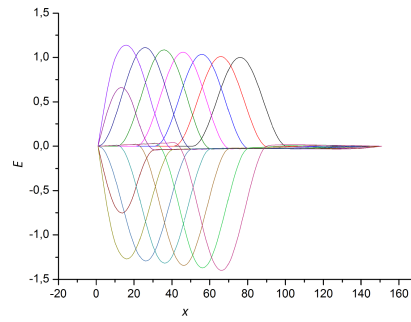
$$\mathbf{F}_1^e(1) = \frac{\Delta t^2(2B^0 - 2B^1 + h(J^0 + J^1))}{8h} + \frac{1}{6}h(B^0 + 2B^1) - \frac{E^0 \Delta t}{2} \quad (3.46)$$

$$\begin{aligned} \mathbf{F}_1^e(2) = & \frac{1}{24h} \{2h\Delta t((\xi^0 + 1)E^0 + 2(\xi^1 + 1)E^1) + \\ & + \Delta t^2(-(3B^0 - 3B^1 + h(J^0 + 2J^1))) + 4h(J^0 + 2J^1)\} \end{aligned} \quad (3.47)$$

Pomocí programu 3.4 jsem modeloval různé situace. Nejprve porovnáme přenos pulzu při nulové disperzi ( $\xi = 0$ ) pro různé konstanty  $\Delta t$  (obr. 3.5) a srovnáme s výstupem programu, který počítá vlnu bez  $j$  (to odpovídá vynechání poslední rovnice a vynulování pravé strany v rovnici 3.38) na obr. 3.6. Ve všech případech jsem počítal s  $h = 1$ . Vidíme, že s přidáním další rovnice do soustavy se zkracuje maximální možný časový krok. Na základě získaných dat jsem dále používal pro časovou diskretizaci krok  $\Delta t = 0,001$ . Dále jsem situaci modifikoval tak, že  $\xi \neq 0$ ,  $x \in (100, 200)$ . Na obrázku 3.7 jsem volil  $\xi = 0,1$ . Vidíme, že vlna se prakticky celá odrazila, což lze interpretovat tak, že plasmová frekvence prostředí byla vyšší, než frekvence pulzu. To je nakonec jasné už z porovnání obrázku, definičního vztahu pro  $\xi$  a  $\omega_0$  (3.32). Potom jsem volil  $\xi = 0,01$ . Na obr. 3.8 vidíme, že alespoň část pulzu prošla bariérou a po opuštění prostředí s nenulovým  $\xi$  si zachovala amplitudu. Nakonec jsem vyslal dva pulzy nacházející se v prostředí s  $\xi = 0,001$  a  $\xi = 0,0001$  (obr. 3.9).

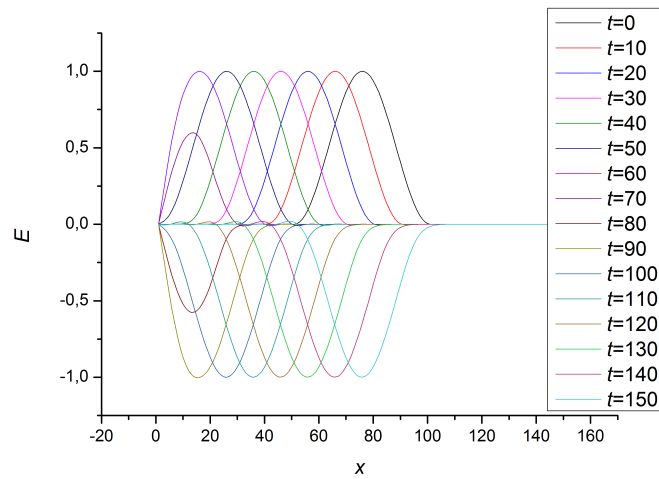


(a)  $\Delta t = 0,001$

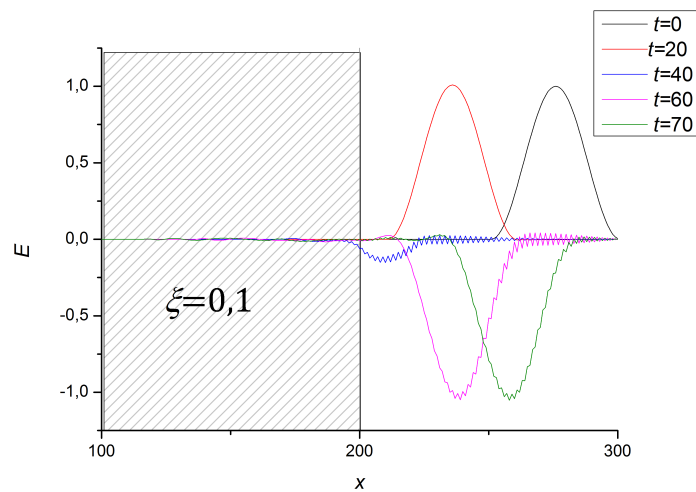


(b)  $\Delta t = 0,01$

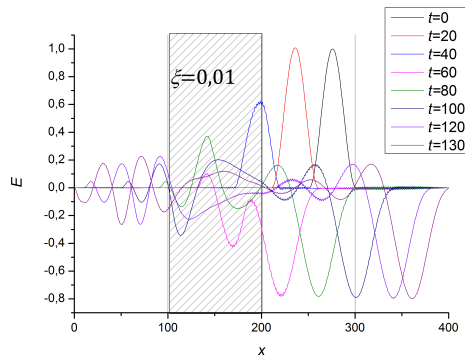
Obrázek 3.5: Porovnání vlny v prostředí bez disperze počítané podle schématu 3.38 pro různé hodnoty  $\Delta t$ . Vlna se pohybuje doleva, vlevo je Dirichletova okrajová podmínka  $E(0, t) = 0$  a černá barva vyznačuje počáteční podmínku.



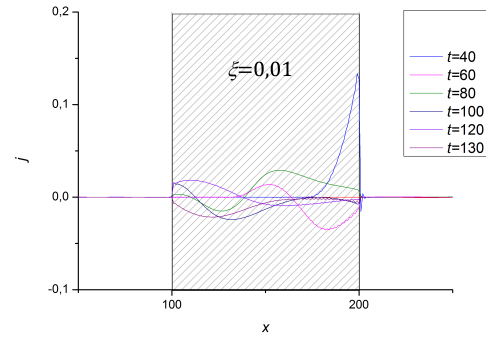
Obrázek 3.6: Vlna počítaná zjednodušeným schématem bez proudové hustoty s  $\Delta t = 1, 0$



Obrázek 3.7: EM pulz dopadající na bariéru s  $\xi = 0,01$

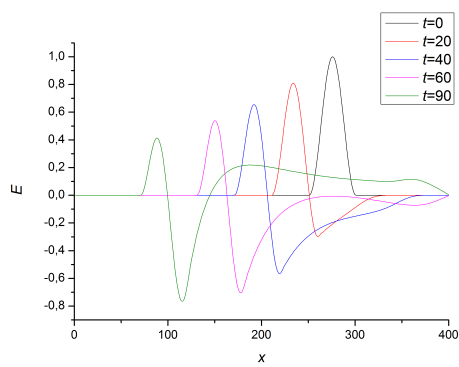


(a) Vývoj elektrické intenzity

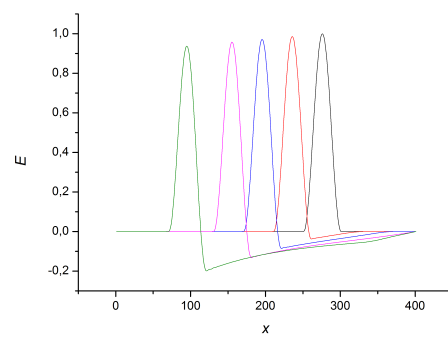


(b) Vývoj proudové hustoty

Obrázek 3.8: EM pulz dopadající na barieru s  $\xi = 0,01$



(a)  $\xi = 0,001$



(b)  $\xi = 0,0001$

Obrázek 3.9: Šíření pulzu v prostředí s disperzí

# Závěr

V průběhu řešení této práce se po zvládnutí teorie shrnuté v prvních dvou kapitolách podařilo odvodit schémata pro numerické řešení eliptické a parabolické rovnice, stejně jako pro hyperbolickou soustavu tří diferenciálních rovnic. Ve všech třech případech byla k řešení v jednodimenzionální prostorové oblasti použita metoda konečných prvků. Poissonova rovnice byla formulována pomocí soustavy dvou rovnic tak, aby na ní mohla být snadno aplikována metoda LSFEM. Po vytvoření programu byla numerická řešení porovnána s analytickým a ukázala se dobrá shoda i při použití pouze několika uzlů. Časově závislé problémy byly v časové oblasti diskretizovány pomocí metody konečných diferencí. Použitím Galerkinovy metody vážených reziduí na rovnici vedení tepla bylo možné srovnat různé přístupy k řešení diferenciálních rovnic pomocí konečných prvků. Počáteční a okrajové podmínky jsem volil tak, aby bylo řešení možné porovnat s analytickým řešením získaným jak pomocí fundamentálního řešení, tak pomocí metody separace proměnných. V obou případech analytická řešení velmi dobře odpovídala tomu numerickému. Nakonec jsem odvodil z Maxwellových rovnic schéma pro vlnu v elektronovém plazmatu a stejně jako v případě Poissonovy rovnice jsem k jejímu řešení použil metodu LSFEM. Ukázal jsem, jaký vliv má přidání třetí rovnice na stabilitu řešení. Soustavu jsem řešil pro fyzikálně zajímavé případy elektromagnetického pulzu dopadajícího na bariéru a pulzu šířícího se v plazmatu. Právě na příkladu pulzu si mohl čtenář uvědomit obrovské výhody adaptivních metod. Zatímco my jsme používali stejnou délku elementu v oblasti, kde se pulz nacházel, i tam, kde nic nebylo, je jasné, že mnohem rychleji a nejspíše i k přesnějšímu řešení bychom mohli dospět, pokud bychom v okolí pulzu diskretizaci velmi zjemnili (a naopak jinde nepočítali zbytečně uzly, na kterých se hodnoty nemění vůbec). Práce tak slouží k nastínění silné stránky adaptivních metod a k pobídnutí čtenáře k jejich dalšímu studiu. Řešené jednoduché příklady sloužily k ilustraci použití metody FEM pro řešení různých typů PDR. Konečným cílem je ovšem aplikace FEM na soustavy nelineárních rovnic (zejména MHD rovnic) a řešení otevřených otázek současné astrofyziky pomocí pokročilejších numerických simulací.

# Seznam použité literatury

- [1] CHUNG, T.J. *Computational fluid dynamics*. Cambridge University Press, 2006. ISBN 05-215-9416-2.
- [2] EVANS, L.C. *Partial differential equations*. 2nd ed. American Mathematical Society, Providence, 2010. ISBN 08-218-4974-3.
- [3] HOFFMAN, Joe.D. *Numerical methods for engineers and scientists*. 2nd ed. Marcel Dekker, New York and Basel, 2001. ISBN 0-8247-0443-6.
- [4] STRAUSS, Walter.A. *Partial differential equations - an introduction*. Johny Wiley Sons, Hoboken, 2008. ISBN 978-0470-05456-7.
- [5] SANDERSON, Conrad *Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments*. Technical Report, NICTA, 2010. 10.5.2014 dostupné z [http://arma.sourceforge.net/armadillo\\_nicta\\_2010.pdf](http://arma.sourceforge.net/armadillo_nicta_2010.pdf)
- [6] SOLIN, P. et al. *Hermes - Higher-Order Modular Finite Element System (User's Guide)*. 10.5.2014 dostupné z <http://hpfem.org/>
- [7] BENZI, Michelle *Key Moments in the History of Numerical Analysis*. Society for Industrial and Applied Mathematics. 2009. 10.5.2014 dostupné z [http://history.siam.org/pdf/nahist\\_Benzi.pdf](http://history.siam.org/pdf/nahist_Benzi.pdf)

# Seznam použitých zkratek

PDR	Parciální diferenciální rovnice nebo soustava parciálních diferenciálních rovnic
ODR	Obyčejná diferenciální rovnice
FDM	Metoda konečných diferencí
FVM	Metoda konečných objemů
FEM	Metoda konečných prvků
RVT	Rovnice vedení tepla
$h$	délka prvku FEM
$p$	stupeň polynomu sloužící k aproximaci v FEM
$hp$ FEM	FEM které je adaptivní vzhledem k $p$ a $h$
MHD	Magnetohydrodynamika

# Přílohy

Zdrojový kód 3.1: Thomasův algoritmus

```
#include <armadillo>
using namespace arma;

typedef cube::fixed<s,s,N>    diag_vedl; //s: n. of eq. in sys.
typedef cube::fixed<s,s,N+1> diag_hl;
typedef cube::fixed<s,1,N+1> trivektor;

trivektor thomas(diag_vedl a,diag_hl b,diag_vedl c,trivektor d)
{
    trivektor di;
    trivektor ui;
    diag_vedl ci;

    ci.slice(0)=-inv(b.slice(0))*c.slice(0);
    for (int i=1;i<N;i++)
        ci.slice(i)=-inv(b.slice(i)+a.slice(i-1)*ci.slice(i-1))*c.slice(i);
    di.slice(0)=inv(b.slice(0))*d.slice(0);
    for (int i=1;i<(N+1);i++)
        di.slice(i)=inv(b.slice(i)+a.slice(i-1)*ci.slice(i-1))*(d.slice(i)-
        a.slice(i-1)*di.slice(i-1));
    ui.slice(N)=di.slice(N);
    for (int i=(N-1);i>(-1);i--)
        ui.slice(i)=di.slice(i)+ci.slice(i)*ui.slice(i+1);

    return(ui);
}
```

Zdrojový kód 3.2: Poissonova rovnice

```
#include <iostream>
#include <fstream>
#include <armadillo>
#include <cmath>
#define h 2.5 //element size
#define N 2
#define pi 3.141592
using namespace std;
using namespace arma;
typedef cube::fixed<2, 2, N> diag_vedl;
typedef cube::fixed<2, 2, N+1> diag_hl;
typedef cube::fixed<2, 1, N+1> trivektor;

int main(/*int argc, char** argv*/)
```

```

{
diag_vedl a0;
trivektor d0;d0.fill(0);
diag_vedl c0;
diag_hl b0;
trivektor u;
ofstream out; //setting output
out.open("out2.dat");
u.fill(0); //state vector u
for (int i=0;i<N+1;i++) out<<u(0,0,i)<<"_";
out<<'\n';

vec F0(2);
vec F1(2);
for (int i=0;i<(N+1);i++) //stiffness matrix
{
b0(0,0,i)=2/h;
b0(0,1,i)=0;

b0(1,0,i)=0;
b0(1,1,i)=2/h+2*h/3;
}
for (int i=0;i<N;i++)
{
c0(0,0,i)=-1/h;
c0(0,1,i)=0.5;

c0(1,0,i)=-0.5;
c0(1,1,i)=-1/h+h/6;
}
for (int i=0;i<N;i++) a0.slice(i)=c0.slice(i).t();
//B.C
c0(0,0,0)=0;c0(0,1,0)=0; //D.B.C.
b0(0,0,0)=1;
b0(0,1,0)=0;
b0(1,0,0)=0.5;
b0(1,1,0)=1/h+h/3;
b0(1,0,N)=-0.5;
b0(1,1,N)=1/h+h/3;
b0(0,0,N)=1;
b0(0,1,N)=0;
a0(0,0,N-1)=0;a0(0,1,N-1)=0;
for (int i=0;i<N;i++)
{
F0(0)=0;
F0(1)=-1;
F1(0)=0;

```



```

        F1(1)=+1;
        d0.slice(i)=d0.slice(i)+F0;
        d0.slice(i+1)=d0.slice(i+1)+F1;
    }
    d0(0,0,0)=-3;d0(0,0,N)=1;

    u=thomas(a0,b0,c0,d0); //solve

    //output
    std::cout<<b0(1,0,0)<<"<Initial>"<<'/'<<'/'<<a0<<b0<<c0;
    for (int i=0;i<N+1;i++) out<<u(0,0,i)<<"_";
        out<<'\\n';

    out.close();
return 0;
}

```

### Zdrojový kód 3.3: Rovnice vedení tepla

```

#include <iostream>
#include <fstream>
#include <armadillo>
#include <cmath>
#define h 1.0 //element size
#define N 200 //number of nodes except for boundaries
#define dt 0.02 //time step
#define pi 3.141592
#define t_max 100 //max time
#define alpha 1.0 //conductivity
#define uN0 0.0 //BC 0
#define uNN 0.0 //BC N
using namespace std;
using namespace arma;
typedef cube::fixed<3, 3, N> diag_vedl;
typedef cube::fixed<3, 3, N+1> diag_hl;
typedef cube::fixed<3, 1, N+1> trivektor;

int main(/*int argc, char** argv*/)
{
    ofstream myfile;
    myfile.open ("RVT_g.dat");
        //Initialisation
    const double c1=6*alpha*dt/(h*h);
    vec u0(N); for(int i=0;i<(90);i++) u0(i)=0; //start state vector
    for(int i=(90);i<110;i++) u0(i)=20;
    for(int i=(110);i<N;i++) u0(i)=0;
    vec u=u0; //state vector inicialization
    mat T(N,N, fill::zeros); // Stifness matrix
    T.diag().fill(4);
    T.diag(1).fill(1);
}

```

```

T.diag(-1)=T.diag(1);
T(0,0)=1; T(0,1)=0; T(N-1,N-2)=0; T(N-1,N-1)=1;
vec Z(N, fill::zeros); //Load vector

std::cout<<"start_iteration"<<'\n';
double t; //time
int j=0;
for (t=0; t<(t_max+1); t=t+dt)
{
  for (int i=0; i<(N) ;i++) //load vector construction
  {
    if (i == 0) Z(i)=uN0+4*u(i)+u(i+1)+c1*(uN0-2*u(i)+u(i+1));
    else if (i == N-1)
    {
      Z(i)=u(i-1)+4*u(i)+uNN+c1*(u(i-1)-2*u(i)+uNN);
    }
    else
    {
      Z(i)=u(i-1)+4*u(i)+u(i+1)+c1*(u(i-1)-2*u(i)+u(i+1));
    }
  }
  solve(u, T, Z); //find new u
  if ((j%500)==0) //output
  {
    for (int i=0; i<N; i++) myfile<<u(i)<<"_";
    myfile<<'\n';
    std::cout<<"time_"<<t<<'\n';
  }
  j++;
}
myfile.close();
return 0;
}

```

Zdrojový kód 3.4: Vlna v plazmatu

```

#include <iostream>
#include <fstream>
#include <armadillo>
#include <cmath>
#define h 1.0 //element size
#define N 300 //number of elements
#define dt 0.0005 //time step
#define pi 3.141592
#define t_max 200 //max time
using namespace std;
using namespace arma;
typedef cube::fixed<3, 3, N> diag_vedl;
typedef cube::fixed<3, 3, N+1> diag_hl;

```

```

typedef cube::fixed <3, 1, N+1> trivektor;

int main(/*int argc, char** argv*/)
    diag_vedl a0;
    trivektor d0;
    diag_vedl c0;
    diag_hl b0;
    trivektor u;
    ofstream txt; txt.open ("wave_plasma.dat");    //setting output

    //initialisation of state vector
    // u[(E=0,B=1,j=2),(0),(position > i or i+1)]
    u.fill(0);
    vec alpha(N+1);
    //initialisation of constant
    alpha.fill(0.005);
    //stiffness matrix
    for (int i=1;i<(N);i++) //diagonal matrices
    {
        b0(0,0,i)=dt*dt/(2*h)+2*h/3+alpha(i)*alpha(i)*dt*dt*h/12+
            alpha(i+1)*alpha(i+1)*dt*dt*h/12;
        b0(0,1,i)=0;
        b0(0,2,i)=(1.0-alpha(i)/2-alpha(i+1)/2)*h*dt/3;
        b0(1,0,i)=0;
        b0(1,1,i)=dt*dt/(2*h)+2*h/3;
        b0(1,2,i)=0;
        b0(2,0,i)=b0(0,2,i);
        b0(2,1,i)=0;
        b0(2,2,i)=2*h/3+h*dt*dt/6;
    }
    for (int i=0;i<N;i++)    //supradiagonal matrices
    {
        c0(0,0,i)=h/6-dt*dt/(4*h)+(1.0/24)*alpha(i)*alpha(i+1)*h*dt*dt;
        c0(0,1,i)=0;
        c0(0,2,i)=dt*h*(1.0-alpha(i+1))/12.0;
        c0(1,0,i)=0;
        c0(1,1,i)=-dt*dt/(4*h)+h/6;
        c0(1,2,i)=-dt*dt/8;
        c0(2,0,i)=dt*h*(1.0-alpha(i))/12.0;
        c0(2,1,i)=dt*dt/8;
        c0(2,2,i)=h*(1.0+dt*dt/4)/6;
    }
    for (int i=0;i<N;i++)  a0.slice(i)=c0.slice(i).t();    //subdiag

    //Boundary conditions in stiffness matrix (currently dirichlet)
    c0(0,0,0)=0;c0(0,1,0)=0;c0(0,2,0)=0;
    b0(0,0,0)=1.0;
    b0(0,1,0)=0;

```

```

b0(0,2,0)=0;
b0(1,0,0)=dt/2;
b0(1,1,0)=dt*dt/(4*h)+h/3;
b0(1,2,0)=dt*dt/8;
b0(2,0,0)=dt*h/6-alpha(0)*h*dt/6;
b0(2,1,0)=b0(1,2,0);
b0(2,2,0)=h*(1+dt*dt/4)/3;
b0(0,0,N)=1;
b0(0,1,N)=-dt/2;
b0(0,1,N)=0;
b0(0,2,N)=dt*h*(1.0-alpha(N))/6;
b0(0,2,N)=0;
a0(0,0,(N-1))=0;a0(0,1,(N-1))=0;a0(0,2,(N-1))=0;
b0(1,0,N)=b0(0,1,N);
b0(1,1,N)=dt*dt/(4*h)+h/3;
b0(1,2,N)=-dt*dt/8;
b0(2,0,N)=b0(0,2,N);
b0(2,1,N)=b0(1,2,N);
b0(2,2,N)=h*(1.0+dt*dt/4)/3;

std::cout<<"start_iteration"<<'\n';
double t; //time
int j=0;
vec F0(3);
vec F1(3);
for (t=0; t<t_max; t=t+dt)
{
    if (j % 10000 == 0) //output
    {
        for (int i=0; i<N+1; i++) txt<<u(0,0,i)<<"_";
        txt<<'\n';
    }
    d0.fill(0);
    for (int i=0;i<N;i++) //load vector
    {
        F0(0)=-((-4*(2*u(0,0,i)+u(0,0,i+1))*h*h+2*h*(-6*u(1,0,i+1) +
            (1+alpha(i))*h*(2*u(2,0,i)+u(2,0,i+1)))*dt+(6*u(0,0,i)-
            6*u(0,0,i+1) + alpha(i)*(2*alpha(i)*u(0,0,i) +
            alpha(i+1)*u(0,0,i+1))*h*h)*dt*dt)/(24*h);
        F0(1)=(1.0/6)*(2*u(1,0,i) + u(1,0,i+1))*h+(u(0,0,i+1)*dt)/2-
            ((2*u(1,0,i) - 2*u(1,0,i+1) + h*(u(2,0,i) +
            u(2,0,i+1)))*dt*dt)/(8*h);
        F0(2)=(1.0/24)*(4*h*(2*u(2,0,i)+u(2,0,i+1))+2*h*dt*(2*(1+
            alpha(i))*u(0,0,i)+(1+alpha(i+1))*u(0,0,i+1))-(3*
            u(1,0,i)-3*u(1,0,i+1)+h*(2*u(2,0,i)+u(2,0,i+1)))*dt*dt);
        F1(0)=-((-4*(u(0,0,i)+2*u(0,0,i+1))*h*h+2*h*(6*u(1,0,i)+

```

```

(1+alpha(i+1))*h*(u(2,0,i)+2*u(2,0,i+1))*dt+
(-6*u(0,0,i)+6*u(0,0,i+1)+alpha(i+1)*(alpha(i)*u(0,0,i)
+ 2*alpha(i+1)*u(0,0,i+1))*h*h)*dt*dt)/(24*h));
F1(1)=(1.0/6)*(u(1,0,i) + 2*u(1,0,i+1))*h-(u(0,0,i)*dt)/2+
((2*u(1,0,i)-2*u(1,0,i+1) + h*(u(2,0,i)+
u(2,0,i+1)))*dt*dt)/(8*h);
F1(2)=(1.0/24)*(4*h*(u(2,0,i) + 2*u(2,0,i+1))+
2*((1 + alpha(i))*u(0,0,i) +
2*(1 + alpha(i+1))*u(0,0,i+1))*h*dt-(3*u(1,0,i)-
3*u(1,0,i+1) + h*(u(2,0,i) + 2*u(2,0,i+1)))*dt*dt);

d0.slice(i)=d0.slice(i)+F0;
d0.slice(i+1)=d0.slice(i+1)+F1;

}
if (t<22) d0(0,0,0)=sin(t/7)*sin(t/7);
else d0(0,0,0)=0; d0(0,0,N)=0; //B.C

u=thomas(a0,b0,c0,d0); // "solve" using thomas()

j++;
}
txt.close();
return 0;
}

```