



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

INVERZNÍ PROBLÉMY V ÚLOHÁCH PŘENOSU TEPLA S FÁZOVÝMI ZMĚNAMI

INVERSE PROBLEMS IN COMPUTATIONAL HEAT TRANSFER WITH PHASE CHANGE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Kamarýt

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Lubomír Klimeš, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav: Ústav matematiky
Student: **Bc. Petr Kamarýt**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **doc. Ing. Lubomír Klimeš, Ph.D.**
Akademický rok: 2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Inverzní problémy v úlohách přenosu tepla s fázovými změnami

Stručná charakteristika problematiky úkolu:

V mnoha úlohách technické praxe je zapotřebí zpětného stanovení počátečních/okrajových podmínek či vlastností materiálů na základě znalosti o chování a vývoji daného systému v čase. Jednou z těchto oblastí je i výpočtové řešení úloh přenosu tepla. Vytvoření vhodného výpočtového systému pro řešení těchto problémů je tedy užitečným nástrojem pro analýzu těchto problémů a jejich následné využití v praktických aplikacích.

Cíle diplomové práce:

Cílem práce je seznámit se s problematikou inverzních úloh přenosu tepla, vytvořit numerický model přenosu tepla vhodného systému zahrnujícího fázovou změnu, implementovat zvolené matematické metody pro řešení inverzní úlohy a tyto metody otestovat na testovacích úlohách a vzájemně porovnat jejich chování.

Seznam doporučené literatury:

INCROPERA, F. P., DEWITT, D. P., BERGMAN, T. L. a LAVINE, A. S. Principles of heat and mass transfer. Sedmé vydání. Singapore: John Wiley, 2013. ISBN 978-0-470-64615-1.

HAHN, D. W. a ÖZİŞIK, M. N. Heat conduction. Třetí vydání. Hoboken: Wiley, 2012. ISBN 978-0470902936.

ALIFANOV, M. O. Inverse Heat Transfer Problems. První vydání. Berlín: Springer-Verlag, 1994. ISBN 978-3-642-76438-7.

CHANG, M.-H. Development and Applications of Inverse Heat Transfer Method. První vydání. Riga: VDM Verlag, 2009. ISBN 978-3639140934.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato diplomová práce se zabývá inverzními problémy v úlohách přenosu tepla se změnou fáze. V první kapitole jsou popsány mechanismy přenosu tepla včetně modelování materiálu se změnou fáze. Druhá kapitola se věnuje výpočtovému řešení úloh přenosu tepla. Ve třetí je formulována inverzní úloha pro neznámou okrajovou podmínku. Čtvrtá kapitola popisuje autorem implementované metody pro řešení přímých a inverzních úloh přenosu tepla. Inverzní úlohy jsou řešeny sekvenční metodou a pomocí umělých neuronových sítí. Byly zvoleny dva průběhy hustoty tepelného toku: spojitý, po částech lineární a nespojitý, po částech konstantní. Obě metody dosahují pro obě úlohy srovnatelných výsledků. V případě nespojitě hustoty tepelného toku jsou výsledky horší než v případě spojitěho průběhu.

Abstract

This diploma thesis deals with inverse problems in heat transfer with phase change. The first chapter focuses on heat transfer mechanisms including phase change. Second chapter deals with computational solution of heat transfer problems. In the third chapter the inverse problem for heat flux estimation is formulated. Fourth chapter is description of methods, implemented by the author, for computational solution of both direct and inverse heat transfer problems. Solution of inverse problem are obtained by the sequential method and artificial neural networks. Two heat flux types were selected: continuous, piecewise linear and discontinuous, piecewise constant. Obtained result for both cases are comparable. In case of discontinuous heat flux, results are worse than in continuous case.

Klíčová slova

inverzní úlohy, přenos tepla se změnou fáze, sekvenční metoda, neuronové sítě

Keywords

inverse problems, heat transfer with phase change, sequential method, neural networks

Citace

KAMARÝT, Petr. *Inverzní problémy v úlohách přenosu tepla s fázovými změnami* [online]. Brno, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/141047>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematiky. Vedoucí práce Lubomír Klimeš.

Prohlášení

Prohlašuji, že jsem diplomovou práci *Inverzní problémy v úlohách přenosu tepla s fázovými změnami* vypracoval samostatně pod vedením doc. Ing. Lubomíra Klimeše, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Petr Kamarýt
20. května 2022

Podpora

Tato diplomová práce vznikla za podpory projektu 22-31173S Adaptivní soft computing framework pro řešení inverzních úloh přenosu tepla se změnou skupenství.

Poděkování

Chtěl bych poděkovat doc. Ing. Lubomíru Klimešovi, Ph.D. za vedení mé diplomové práce, ochotu a čas, který jí věnoval.

Obsah

Úvod	8
1 Přenos tepla	9
1.1 Vedení tepla	9
1.2 Konvekce	13
1.3 Radiace	13
1.4 Přímá úloha vedení tepla	13
1.5 Změna fáze	15
2 Numerické metody řešení přímých úloh vedení tepla	19
2.1 Eliptické úlohy	19
2.2 Parabolické úlohy	25
2.3 Nelineární úlohy	27
3 Inverzní úlohy přenosu tepla	28
3.1 Formulace inverzní úlohy vedení tepla	28
3.2 Metoda odhadu funkce	30
3.3 Umělé neuronové sítě	32
4 Implementace metod	35
4.1 Funkce řešící přímou úlohu vedení tepla	35
4.2 Funkce řešící inverzní úlohu vedení tepla	44
5 Dosažené výsledky a porovnání metod	48
5.1 Řešená inverzní úloha	48
5.2 Výsledky a porovnání metod	50
Závěr	52
Seznam použitých zdrojů	54
Seznam symbolů	56
Seznam obrázků	58

Úvod

Inverzní úlohy přenosu tepla hrají důležitou roli v technické praxi. Rozvoj inverzních úloh se datuje do roku 1960, ve kterém byl G. Stolzem publikován jeden z prvních článků na toto téma. Stolz řešil praktické úlohy přenosu tepla při kalení součástí jednoduchých tvarů. Další oblasti, ve kterých se vyskytla potřeba řešit inverzní úlohy, jsou kosmický program, testování součástí jaderných reaktorů, tuhnutí skla a slévárenství.

Přímou úlohou bude myšleno stanovení vývoje systému na základě počátečních a okrajových podmínek a fyzikálních vlastností materiálu. Inverzní úlohou pak bude myšlena úloha zpětného stanovení těchto podmínek a vlastností ze znalosti vývoje systému. Inverzní úlohy patří z matematického hlediska mezi špatně podmíněné úlohy, což znamená, že často nejsou splněny předpoklady pro existenci a jednoznačnost jejich řešení. Nalézt toto řešení je tedy obtížnější než nalézt řešení přímé úlohy.

Cílem této diplomové práce je seznámit se s problematikou inverzních úloh přenosu tepla, vytvořit numerický model přenosu tepla vhodného systému s fázovou změnou, implementovat zvolené matematické metody pro řešení inverzní úlohy a tyto metody aplikovat na testovací úlohy a vzájemně porovnat jejich chování.

V první kapitole jsou rozebrány mechanismy přenosu tepla, včetně fázové změny, což umožní formulaci matematického modelu systému se změnou fáze. Přímé úlohy, zejména s fázovou změnou, jsou analyticky řešitelné spíše výjimečně. Inverzních úlohy vyžadují řešení velkého množství úloh přímých, proto se další kapitola zabývá numerickými metodami pro přímé úlohy. Třetí kapitola je věnována formulaci inverzní úlohy pro neznámou okrajovou podmínku (hustotu tepelného toku) a popisem autorem zvolených metod řešení této úlohy.

Čtvrtá kapitola se zaměřuje na autorem implementované funkce pro numerický výpočet přímých a inverzních úloh. Poslední kapitola je věnována formulaci matematického modelu systému se změnou fáze. Řešením přímých testovacích úloh je získán vývoj teploty tohoto systému v čase, což je následně použito jako vstupní data pro metody, které řeší inverzní úlohu. V závěru kapitoly jsou dosažené výsledky zhodnoceny a kvantifikovány.

Kapitola 1

Přenos tepla

Přenos tepla v tělese probíhá obvykle třemi mechanizmy: vedením (kondukcí), prouděním (konvekcí) a zářením (radiací). Vedení tepla bude pro tuto diplomovou práci stěžejní, jeho popisu tedy bude věnována největší pozornost. Konvekce a radiace se mohou vyskytovat jako okrajové podmínky při řešení úloh vedení tepla, proto budou popsány jen stručně.

1.1 Vedení tepla

Vedení tepla lze popsat jako přenos tepla z míst s vyšší teplotou do míst s nižší teplotou, tento přenos probíhá díky neuspořádanému pohybu částic látky tělesa. Těmito částicemi mohou být molekuly (kapaliny a plyny), nebo atomy (tuhé látky). S vedením tepla se lze nejčastěji setkat u tuhých těles. Podmínkou pro vedení tepla je to, aby v tělese existoval nenulový teplotní gradient, tuto skutečnost popisuje známý Fourierův zákon [7].

Fourierův zákon

Přenos tepla vedením v homogenním, izotropním tělese probíhá podle vztahu

$$\dot{\mathbf{q}}(\mathbf{x}, t) = -\lambda \nabla T(\mathbf{x}, t), \quad (1.1)$$

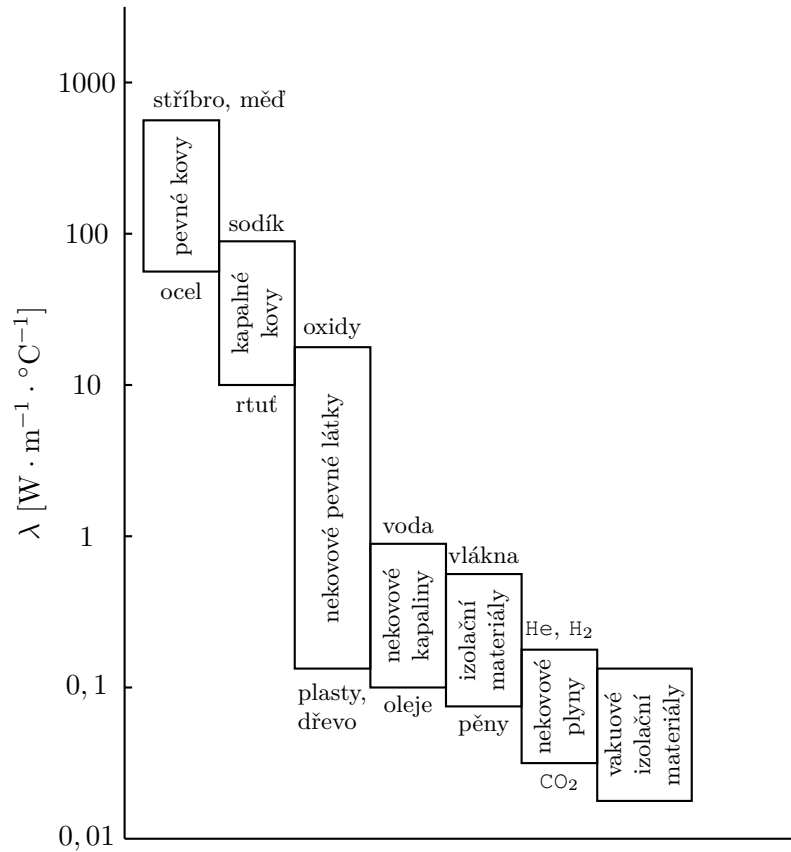
kde $\dot{\mathbf{q}}$ [$\text{W} \cdot \text{m}^{-2}$] je hustota tepelného toku, λ [$\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$] je součinitel tepelné vodivosti a $\nabla T(\mathbf{x}, t)$ [$\text{K} \cdot \text{m}^{-1}$] je teplotní gradient¹. Teplota tělesa $T = T(\mathbf{x}, t)$ [K] je funkcí polohy $\mathbf{x} = (x, y, z)$ a času t .

Vzhledem ke vztahu (1.1) a interpretaci gradientu lze také říci, že přenos tepla vedením probíhá ve směru největšího poklesu teploty. Lze také říci, že teplo se v tělese šíří vedením ve směru kolmém na izotermy.

Součinitel tepelné vodivosti

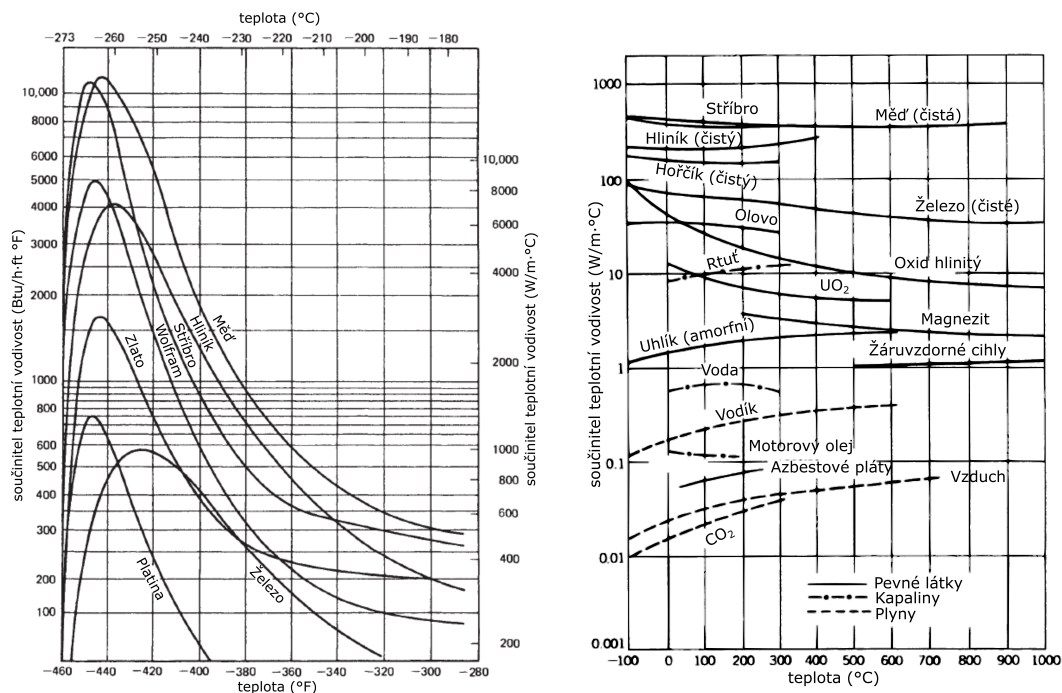
Součinitel tepelné vodivosti λ ve vztahu (1.1) je fyzikální vlastností látky tělesa a může nabývat širokého spektra hodnot. Na obrázku 1.1 jsou hodnoty součinitele tepelné vodivosti pro různé skupiny látek. Obecně lze říci, že nejvyšší hodnoty mají čisté kovy, naopak nejnižší hodnoty mají plyny a páry.

¹Teplotní gradient je v kartézské souřadné soustavě definován jako $\nabla T(\mathbf{x}, t) = \left(\frac{\partial T(\mathbf{x}, t)}{\partial x}, \frac{\partial T(\mathbf{x}, t)}{\partial y}, \frac{\partial T(\mathbf{x}, t)}{\partial z} \right)^\top$.



Obrázek 1.1: Hodnoty součinitele tepelné vodivosti různých látek podle [7]

Součinitel tepelné vodivosti není konstantní, u většiny látek závisí na teplotě. Konstantní tepelnou vodivost v matematickém modelu vedení tepla lze předpokládat, pokud bude model omezen na malý interval teplot. Pro většinu čistých kovů se tepelná vodivost s rostoucí teplotou snižuje, zatímco pro plyny se s rostoucí teplotou zvyšuje. Na obrázku 1.2 (vpravo) je ukázána závislost tepelné vodivosti na teplotě pro vybrané látky. Tepelná vodivost kovů při teplotách blízkých absolutní nule se blíží nule, jak lze vidět na obrázku 1.2 (vlevo).[7]



Obrázek 1.2: Závislost součinitele tepelné vodivosti na teplotě [7] (u obou obrázků byly přeloženy popisky)

Existují také látky, u nichž se hodnota tepelné vodivosti liší v závislosti na směru vedení tepla. Jako příklad takové látky lze uvést grafit, zde se hodnota tepelné vodivosti mezi dvěma směry liší o dva řády [7]. Látky s touto vlastností se nazývají anizotropní. Jestliže se hodnoty tepelné vodivosti liší ve vzájemně kolmých směrech, pak se tato látka nazývá ortotropní. Dále je předpokládáno, že látka je izotropní, tedy že má tepelnou vodivost nezávislou na směru vedení tepla.

Odvození rovnice vedení tepla

V následujících odstavcích bude odvozena rovnice vedení tepla v trojrozměrném tělese. K odvození bude využit zákon zachování energie pro kontrolní objem V tělesa $\Omega \subset \mathbb{R}^3$ [7]. Kontrolní objem je zvolen libovolně jako $V \subset \Omega$, pak zákon zachování energie pro V lze psát ve tvaru

$$E_V = Q_g + Q_S, \quad (1.2)$$

kde E_V je energie akumulovaná v objemu V za jednotku času, Q_S teplo, které z objemu V vyteče přes jeho hranici S , za jednotku času a Q_g teplo, vygenerované v objemu V vnitřními zdroji tepla s hustotou výkonu $g = g(\mathbf{x}, t)$ [$\text{W} \cdot \text{m}^{-3}$], za jednotku času.

Nyní budou vyjádřeny jednotlivé veličiny ze zákona zachování energie jako integrály přes V . Teplo vygenerované vnitřními zdroji je vyjádřeno jako

$$Q_g = \int_V g(\mathbf{x}, t) \, dx. \quad (1.3)$$

Teplo, které z V vyteče přes hranici ∂V objemu V je vyjádřeno pomocí hustoty tepelného toku $\dot{\mathbf{q}}$ takto

$$Q_S = - \int_{\partial V} \dot{\mathbf{q}} \cdot \mathbf{n} \, dS = - \int_V \nabla \cdot \dot{\mathbf{q}} \, dx, \quad (1.4)$$

kde \mathbf{n} je vnější normála k hranici S a $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^\top$ je gradient. V poslední rovnosti vztahu (1.4) byla využita Gaussova-Ostrogradského věta, pomocí které byl plošný integrál přes hranici ∂V převeden na trojný integrál přes kontrolní objem V . energii akumulovanou v objemu V lze, podle [7], vyjádřit pomocí vztahu

$$E_V = \int_V \rho c \frac{\partial T}{\partial t} d\mathbf{x}, \quad (1.5)$$

kde c [$\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$] je fyzikální veličina nazývaná měrná tepelná kapacita a ρ [$\text{kg} \cdot \text{m}^{-3}$] je hustota. Nyní budou vztahy (1.3), (1.4) a (1.5) dosazeny do zákona zachování energie (1.2), čímž je obdržěn vztah

$$\int_V \left(\rho c \frac{\partial T}{\partial t} - g + \nabla \cdot \dot{\mathbf{q}} \right) d\mathbf{x} = 0.$$

Dále je využit Fourierův zákon (1.1)

$$\int_V \left(\rho c \frac{\partial T}{\partial t} - g - \nabla \cdot (\lambda \nabla T) \right) d\mathbf{x} = 0.$$

Protože kontrolní objem V je libovolný, lze psát

$$\rho c \frac{\partial T}{\partial t} - g - \nabla \cdot (\lambda \nabla T) = 0.$$

Výše uvedená rovnice bývá častěji zapisována ve tvaru

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \nabla T) + g.$$

Tato rovnice se nazývá (nestacionární) rovnice vedení tepla v tělese, neznámou je funkce $T = T(\mathbf{x}, t)$. Jestliže je navíc tepelná vodivost konstantní, pak lze rovnici vedení tepla zjednodušit na tvar

$$\rho c \frac{\partial T}{\partial t} = \lambda \Delta T + g,$$

kde ΔT je Laplaceův operátor². Vydělením rovnice součinitelem tepelné vodivosti lze dostat rovnici

$$\frac{1}{a} \frac{\partial T}{\partial t} = \Delta T + \frac{g}{\lambda},$$

kde $a = \frac{\lambda}{\rho c}$ [$\text{m}^2 \cdot \text{s}^{-1}$] je fyzikální veličina, která se nazývá součinitel teplotní vodivosti. Pokud je navíc uvažován ustálený stav vedení tepla (tedy $\frac{\partial T}{\partial t} = 0$), pak lze psát stacionární rovnici vedení tepla

$$-\lambda \Delta T = g.$$

Neznámá T je v tomto případě pouze funkce polohy, tedy $T = T(\mathbf{x})$.

Všechny výše uvedené rovnice lze také transformovat do jiných souřadných systémů, např. sférického, cylindrického apod. V případě dvourozměrného tělesa lze použít polární souřadný systém. Tyto transformované rovnice lze nalézt v [7], zde ovšem uvedeny nebudou. Použitím Fourierova zákona pro ortotropní, resp. anizotropní těleso při odvozování rovnice vedení tepla je možno odvodit rovnice vedení tepla pro ortotropní, resp. anizotropní těleso. Lze také uvažovat různé kombinace výše uvedeným přístupů, např. stacionární vedení tepla v ortotropním materiálu apod.

²Laplaceův operátor je v kartézském souřadné soustavě definován takto $\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}$.

1.2 Konvekce

Přenos tepla konvekcí se vyskytuje při obtékání povrchu pevného tělesa tekutinou. Jedná se o kombinaci dvou jevů: difuze (jako u přenosu tepla vedením) a proudění okolní tekutiny. Tepelný tok z tělesa do tekutiny (případně obráceně) lze popsat pomocí Newtonova ochlazovacího zákona [7].

Newtonův ochlazovací zákon

Velikost hustoty tepelného toku mezi povrchem pevného tělesa obtékaného tekutinou je dána vztahem

$$\dot{q} = h(T - T_\infty),$$

kde h [$\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$] je součinitel přestupu tepla, T_∞ je teplota obklopující tekutiny a T je teplota na povrchu tělesa. Směr hustoty tepelného toku je dán normálou k povrchu obtékaného tělesa.

1.3 Radiace

Každé těleso s teplotou vyšší než 0 K vyzařuje energii. Tento jev se nazývá tepelné záření. Na rozdíl od přenosu tepla vedením a konvekcí, přenos tepla zářením může probíhat i ve vakuu. Tepelné záření tělesa je popsáno Stefanovým-Boltzmannovým zákonem [7].

Stefanův-Boltzmannův zákon

Velikost hustoty zářivého tepelného toku vyzářeného z povrchu tělesa je dána vztahem

$$\dot{q} = \varepsilon\sigma T^4,$$

kde ε je poměrná zářivost, σ je Stefanova-Boltzmannova konstanta $\sigma = 5,669 \cdot 10^{-8}$ [$\text{W} \cdot \text{m}^{-2} \cdot \text{K}^4$] a T je teplota povrchu tělesa (v kelvinech). Velikost hustoty tepelného toku mezi povrchem tělesa a obklopujícím médiem je možné, podle [7], aproximovat vztahem

$$\dot{q} = \varepsilon\sigma(T^4 - T_\infty^4),$$

kde T_∞ je teplota obklopujícího média. Tento vztah platí, jestliže je povrch tělesa obklopen mnohem větším povrchem, např. v situaci, kdy malý zdroj tepla vyzařuje do velké místnosti [14].

1.4 Přímá úloha vedení tepla

Nestacionární rovnice vedení tepla je parciální diferenciální rovnice parabolického typu, stacionární rovnice je typu eliptického. V obou případech je tedy pro řešení rovnice nutno zadat okrajové podmínky na hranici oblasti Ω . V nestacionárním případě je nutno zadat také počáteční podmínku (počáteční teplotu). Rovnice vedení tepla daná na oblasti Ω (a intervalu (t_0, t_f) v nestacionárním případě), okrajové podmínky a případně počáteční podmínka pak tvoří přímou úlohu vedení tepla. Jejím řešením je funkce polohy a času $T = T(\mathbf{x}, t)$, která popisuje vývoj teploty v tělese. Počáteční podmínku je zadána v každém bodě tělesa Ω

$$T(\mathbf{x}, t_0) = T_0(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

kde funkce $T_0(\mathbf{x})$ je dána. Budou rozlišovány tři typy okrajových podmínek: Dirichletova, Neumannova a Newtonova [7]. Okrajové podmínky se předepisují na hranici oblasti Ω , tato hranice je označena $\Gamma = \partial\Omega$, dále \mathbf{n} označuje jednotkový vektor vnější normály ke Γ . Dirichletova okrajová podmínka je tvaru

$$T(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \quad t \in (t_0, t_f),$$

kde f je známá funkce. Dirichletova okrajová podmínka tedy předepisuje teplotu na hranici Γ . Neumannova okrajová podmínka předepisuje na hranici Γ hustotu tepelného toku ve směru normály

$$-\lambda \frac{\partial T(\mathbf{x}, t)}{\partial n} = \dot{q}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \quad t \in (t_0, t_f),$$

kde $\frac{\partial T(\mathbf{x}, t)}{\partial n}$ značí derivaci podle normály³ a \dot{q} je známá funkce. Speciálním případem této okrajové podmínky je

$$-\lambda \frac{\partial T(\mathbf{x}, t)}{\partial n} = 0, \quad \mathbf{x} \in \Gamma, \quad t \in (t_0, t_f),$$

což představuje tepelně izolovanou hranici Γ (tepelný tok ve směru normály ke Γ je nulový, tedy hranicí neprochází žádné teplo). Newtonova okrajová podmínka předepisuje závislost tepelného toku na teplotě

$$-\lambda \frac{\partial T(\mathbf{x}, t)}{\partial n} = \alpha(\mathbf{x})T(\mathbf{x}, t) - \beta(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \quad t \in (t_0, t_f),$$

kde α, β jsou dané funkce. Speciální případ Newtonovy okrajové podmínky je situace, kdy kolem hranice Γ proudí tekutina a mezi tělesem a tekutinou probíhá přenos tepla konvekcí, tj. platí

$$-\lambda \frac{\partial T(\mathbf{x}, t)}{\partial n} = h(T(\mathbf{x}, t) - T_\infty(\mathbf{x}, t)), \quad \mathbf{x} \in \Gamma, \quad t \in (t_0, t_f),$$

kde h je součinitel přestupu tepla a T_∞ je teplota tekutiny.

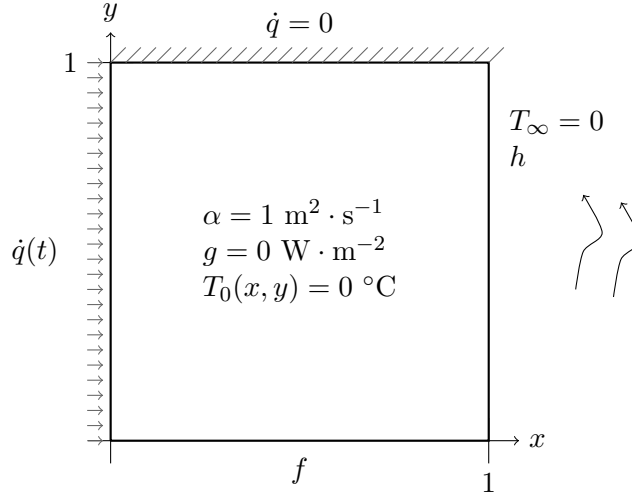
Jestliže některá z fyzikálních vlastností v rovnici vedení tepla ρ, λ, c nebo některá z funkcí α, β (speciálně také h) závisí na teplotě, pak je přímá úloha vedení tepla nelineární. Zejména v těchto případech je hledání řešení klasickými metodami velmi obtížné, často dokonce nemožné. Jak ovšem bude uvedeno v další podkapitole, přímou úlohu vedení tepla se změnou fáze lze formulovat právě jako úlohu nelineární, proto se další kapitola bude věnovat metodám numerickým, které tyto nelineární úlohy umožní snadno vyřešit. V monografii [7] lze nalézt mnoho řešených přímých úloh vedení tepla a také přehled klasických metod, kterými tyto problémy řešit.

Příklad formulace přímé úlohy vedení tepla

Je uvažováno vedení tepla v tenké desce, která má rozměry 1×1 m, počáteční teplota v čase 0 s je 0°C a děj bude sledován do času t_f s. Na levé hraně desky (viz obrázek 1.3) je předepsána velikost hustoty tepelného toku jako funkce času $\dot{q} = \dot{q}(t)$ [$\text{W} \cdot \text{m}^{-2}$], horní hrana je tepelně izolována, pravá hrana je obtékána tekutinou o teplotě $T_\infty = 0^\circ\text{C}$ s koeficientem

³Za jistých předpokladů o funkci T lze také psát $\frac{\partial T}{\partial n} = \nabla T \cdot \mathbf{n}$, pak je Neumannova okrajová podmínka tvaru $-\lambda \nabla T \cdot \mathbf{n} = q$.

přestupu tepla h a na spodní straně je za pomoci termostatu udržována konstantní teplota f . Bude uvažováno nestacionární vedení tepla s jednotkovým součinitelem teplotní vodivosti a izotropní materiál bez tepelných zdrojů.



Obrázek 1.3: Příklad přímé úlohy vedení tepla

Vedení tepla je tedy popsáno rovnicí (viz sekce 1.1)

$$\frac{\partial T(x, y, t)}{\partial t} - \Delta T(x, y, t) = 0, \quad (\mathbf{x}, t) \in [(0, 1) \times (0, 1)] \times (0, t_f),$$

s počáteční podmínkou

$$T(x, y, 0) = T_0(x, y) = 0.$$

Na stranách čtverce je předepsána velikost hustoty tepelného toku, tj. jsou zde předepsány Neumannovy okrajové podmínky, to je zapsáno (je předpokládán také jednotkový součinitel tepelné vodivosti) takto

$$\begin{aligned} -\frac{\partial T(0, y, t)}{\partial n_l} &= \frac{\partial T(0, y, t)}{\partial x} = \dot{q}(t), \quad \mathbf{n}_l = (-1, 0)^\top, \quad y \in (0, 1), \quad t \in (0, t_f), \\ -\frac{\partial T(x, 1, t)}{\partial n_t} &= -\frac{\partial T(x, 0, t)}{\partial y} = 0, \quad \mathbf{n}_t = (0, 1)^\top, \quad x \in (0, 1), \quad t \in (0, t_f), \\ -\frac{\partial T(1, y, t)}{\partial n_r} &= -\frac{\partial T(1, y, t)}{\partial x} = hT(1, y, t), \quad \mathbf{n}_r = (1, 0)^\top, \quad y \in (0, 1), \quad t \in (0, t_f), \\ T(x, 0, t) &= f, \quad x \in (0, 1), \quad t \in (0, t_f). \end{aligned}$$

Úkolem tedy je najít funkci $T(x, y, t)$, která splňuje rovnici vedení tepla, počáteční a okrajové podmínky. Zbývá ještě poznamenat, že tato rovnice vedení tepla pro dvojrozměrné těleso předpokládá, že nedochází k tepelným ztrátám „v ploše“, tedy že na horní a spodní straně tenké desky je dokonalá tepelná izolace.

1.5 Změna fáze

Přímé úlohy vedení tepla se změnou fáze (je uvažováno pouze tání a tuhnutí), jsou důležité v mnoha inženýrských aplikacích. Jako příklady lze uvést výrobu ledu, mražení potravin,

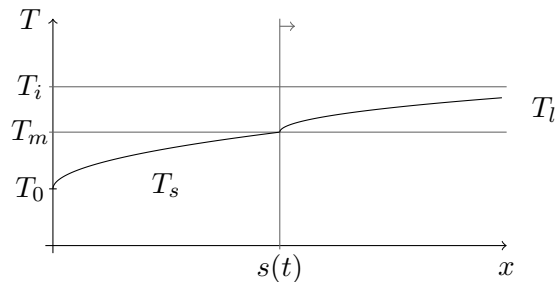
tuhnutí kovů při odlévání, ukládání tepla do tepelných zásobníků, zpracování chemikálií a plastů, růst krystalů, slévání a svařování kovů a mnoho dalších [7]. Řešení těchto problémů je obtížné, protože rozhraní mezi pevnou a kapalnou fází⁴ se v tělese pohybuje a jeho poloha v čase není dopředu známa.

Při změně fáze čistých látek (např. voda), se změna fáze odehrává pouze při teplotě změny fáze, která bude označena T_m . Rozhraní mezi pevnou a kapalnou fází je ostré. Změna fáze směsí, slitin a nečistých látek se odehrává v intervalu teplot, krajní body tohoto intervalu budou označeny T_1 , T_2 . Pevná a kapalná fáze jsou odděleny oblastí, ve které jsou přítomny obě fáze (v anglické literatuře označované jako „mushy“ oblast [7]).

V této kapitole bude nejprve uvedena formulace klasické úlohy přenosu tepla se změnou fáze tzv. Stefanův problém, ve kterém je předpokladem, že tuhnutí probíhá právě při teplotě T_m . Dále budou uvedeny dvě metody, které umožní formulovat problém se změnou fáze do formy, která je vhodná pro numerický výpočet řešení, jedná se o metodu efektivní tepelné kapacity a metodu entalpie.

Stefanův problém

Úloha je řešena na polorovině, ve kterém se nachází tekutina, která má teplotu tuhnutí rovnou teplotě tání T_m . V čase $t = 0$ má tekutina teplotu $T_i > T_m$, dále je v tomto čase na levém konci $x = 0$ skokově snížena teplota na teplotu $T_0 < T_m$, která je zde udržována po celou dobu děje. Kapalina začne tuhnout od levého konce, rozhraní mezi pevnou a kapalnou fází se pohybuje v kladném směru osy x a je označeno $s = s(t)$, jak je znázorněno na obrázku 1.4.



Obrázek 1.4: Tuhnutí kapaliny

Vývoj teploty pevné fáze T_s a kapalně fáze T_l je dán rovnicemi vedení tepla

$$\rho_s c_s \frac{\partial T_s}{\partial t} = \frac{\partial}{\partial x} \left(\lambda_s \frac{\partial T_s}{\partial x} \right), \quad 0 < x < s(t), \quad t > 0,$$

$$\rho_l c_l \frac{\partial T_l}{\partial t} = \frac{\partial}{\partial x} \left(\lambda_l \frac{\partial T_l}{\partial x} \right), \quad s(t) < x < \infty, \quad t > 0.$$

Na rozhraní mezi pevnou a kapalnou fází jsou předepsány dvě podmínky. První z nich vyjadřuje spojitost teploty

$$T_s(s(t), t) = T_l(s(t), t) = T_m, \quad t > 0.$$

⁴Pro označení pevné fáze bude použit písmeno s , podobně pro označení kapalně fáze písmeno l . Vyskytně-li se tedy např. v rovnici fyzikální vlastnost s dolním indexem s , jedná se o vlastnost pevné fáze.

Dále na rozhraní musí platit rovnost mezi teplem odevzdaným kapalnou fází, přijatým pevnou fází a latentním teplem fázové změny, tedy

$$\lambda_s \frac{\partial T(s(t), t)}{\partial x} = \lambda_l \frac{\partial T_l}{\partial x} + \rho L \frac{ds(t)}{dt},$$

kde L [$\text{J} \cdot \text{kg}^{-1}$] je latentní teplo fázové změny. Tyto podmínky se podle [3] nazývají Stefanovy podmínky, důležitým předpokladem pro platnost druhé z nich je, že na rozhraní mezi fázemi nejsou žádné zdroje tepla. Dále je zanedbána změna hustoty v důsledku fázové změny, tedy $\rho_s = \rho_l = \rho$. V úloze se tedy vyskytují tři neznámé T_s, T_l, s , dvě podmínky na rozhraní a dvě okrajové podmínky

$$\begin{aligned} T_s(0, t) &= T_0, \quad t > 0, \\ \lim_{x \rightarrow \infty} T(x, t) &= T_i, \quad t > 0. \end{aligned}$$

Počáteční teplota je předepsána pro kapalnou fází

$$T_l(x, 0) = T_i, \quad x > 0.$$

Tento problém lze řešit hledáním tzv. „similarity solutions“ viz [7] a [3]. Podobně lze formulovat také úlohu pro tání pevné látky. Stefanův problém lze také zobecnit do více dimenzí.

Metoda efektivní tepelné kapacity

Metoda efektivní tepelné kapacity je velmi častou metodou používanou pro modelování úloh vedení tepla se změnou fáze [13]. Předpokladem je, že ke změně fáze dochází v intervalu teplot $\langle T_1, T_2 \rangle$. Uvnitř tohoto intervalu se nejedná ani o kapalnou ani o pevnou fází, jde o přechodovou oblast. Tepelná kapacita látky v intervalu $\langle T_1, T_2 \rangle$ je přímo úměrná latentnímu teplu fázové změny a nepřímo úměrná velikosti tohoto intervalu. Tato tepelná kapacita se nazývá efektivní, tedy

$$c_{\text{eff}} = \frac{L}{T_2 - T_1} + c_s, \quad T \in \langle T_1, T_2 \rangle.$$

Tepelná kapacita látky tělesa je pak definována takto

$$c = \begin{cases} c_s & T < T_1 \\ c_{\text{eff}} & T_1 \leq T \leq T_2 \\ c_l & T > T_2 \end{cases}$$

Takto definovaná tepelná kapacita je pak použita v rovnici vedení tepla, tím se přímá úloha vedení tepla stává nelineární.

Při praktických výpočtech se často tepelná kapacita modeluje funkcí [8]

$$c(T) = c_0 + c_m e^{-\frac{(T-T_m)^2}{\sigma}},$$

kde c_0 je tepelná kapacita vně intervalu teplot fázové změny, c_m je nárůst tepelné kapacity související s latentním teplem fázové změny, T_m je průměrná teplota změny fáze a σ je parametr charakterizující šíři intervalu teplot fázové změny.

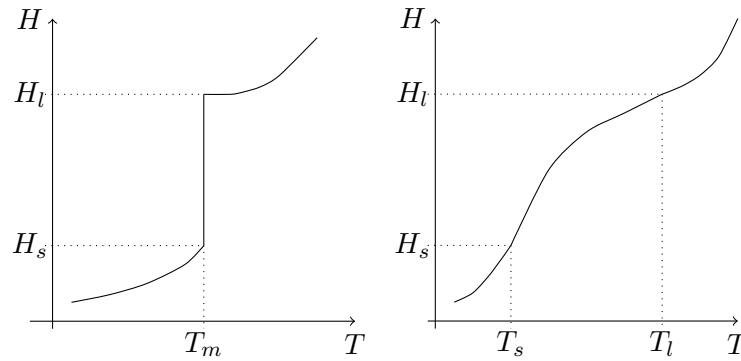
Metoda entalpie

V předchozích dvou formulacích byla jako neznámá u obou fází použita teplota, v metodě entalpie tomu však bude jinak. V úloze se změnou fáze lze vyjádřit zákon zachování energie pro kontrolní objem pomocí entalpie

$$\rho \frac{\partial H(T)}{\partial t} = \nabla \cdot (\lambda \nabla T),$$

kde $H = H(T)$ [$\text{J} \cdot \text{kg}^{-1}$] je měrná entalpie. Lze dokázat [7], že tato rovnice spolu se vztahem mezi entalpií a teplotou je ekvivalentní formulaci úlohy pomocí rovnic vedení tepla pro každou fázi a Stefanových podmínek na rozhraní.

Na obrázku 1.5 lze vidět vztah mezi entalpií a teplotou pro látku, které mění fázi při teplotě T_m (vlevo) a pro látku měnící fázi na intervalu teplot $\langle T_1, T_2 \rangle$ (vpravo).



Obrázek 1.5: Závislost entalpie na teplotě

Podle [7] lze pro látku se změnou fáze při teplotě T_m , vyjádřit závislost entalpie na teplotě ve tvaru

$$H = \begin{cases} c(T - T_m) & T < T_m, \\ c(T - T_m) + L & T > T_m. \end{cases}$$

Naopak, odpovídající závislost teploty na měrné entalpii je, podle [7], vyjádřena vztahem

$$T = \begin{cases} T_m + \frac{H}{c} & H < 0, \\ T_m & 0 \leq H \leq L, \\ T_m + \frac{H-L}{c} & H > L. \end{cases}$$

Pro látku, u kterým nastává fázová změna na intervalu $\langle T_1, T_2 \rangle$ lze, opět podle [7], použít vztah

$$H = \begin{cases} cT & T < T_1, \\ cT + \frac{T-T_1}{T_2-T_1} L & T_1 \leq T \leq T_2, \\ cT + L & T > T_2. \end{cases}$$

Výše popsané rovnice lze řešit vhodnou numerickou metodou. Nejprve se vyřeší rovnice pro entalpii a poté se vypočte teplota.

Metoda efektivní tepelné kapacity má výhodu, že formulace úlohy obsahuje pouze jednu rovnici pro obě fáze, avšak tato rovnice je nelineární. Tuto metodu je vhodné použít, jestliže je interval $\langle T_1, T_2 \rangle$ poměrně široký. Metodu entalpie lze použít, jestliže ke změně fáze dochází při teplotě T_m a také, jestliže ke změně fáze dochází na intervalu $\langle T_1, T_2 \rangle$. Řešení úlohy však probíhá ve dvou krocích.

Kapitola 2

Numerické metody řešení přímých úloh vedení tepla

Řešit přímou úlohu vedení tepla klasickými metodami lze jen v jednoduchých případech, které jsou v praxi spíše výjimečné. Protože řešení inverzních úloh vedení tepla vyžaduje opakované řešení těchto úloh, budou v následující kapitole popsány numerické metody, které tyto úlohy řeší přibližně. Rovnice vedení tepla jsou parciální diferenciální rovnice, které jsou buď parabolického typu v nestacionárním případě, nebo eliptického typu v případě stacionárního vedení tepla. Jelikož úlohy se změnou fáze lze formulovat jako úlohy nelineární (metoda efektivní tepelné kapacity) bude se poslední část této kapitoly věnovat také nelineárním úlohám.

2.1 Eliptické úlohy

Pro řešení okrajových úloh eliptického typu existuje celá řada numerických metod. Mezi ty nejznámější patří metoda konečných diferencí (také metoda sítí), metoda konečných objemů a metoda konečných prvků. Pro řešení přímých úloh vedení tepla byla autorem práce použita metoda konečných prvků, další kapitola bude tedy věnována této metodě, popis metody a praktické rady lze čerpat z [4] a z [5].

Formulace úlohy

Nechť $\Omega \subset \mathbb{R}^2$ je ohraničená oblast, její hranice je označena Γ . Tato hranice se skládá ze dvou částí Γ_1, Γ_2 ¹. Dále $\mathbf{n} = (n_1, n_2)^\top$ značí jednotkovou vnější normálu k této hranici. Úlohou je aproximovat neznámou funkci $u = u(x, y)$ splňující rovnici

$$-\nabla \cdot (p\nabla u) + qu = g, \mathbf{x} \in \Omega, \quad (2.1)$$

na Γ_1 splňující Dirichletovu okrajovou podmínku

$$u = f, \mathbf{x} \in \Gamma_1, \quad (2.2)$$

na Γ_2 splňující Newtonovu² okrajovou podmínku

$$-p \frac{\partial u}{\partial n} = -p\nabla u \cdot \mathbf{n} = \alpha u - \beta, \mathbf{x} \in \Gamma_2, \quad (2.3)$$

¹Je předpokládáno, že Γ_1, Γ_2 jsou disjunktní a že sjednocení jejich uzávěrů dá celou hranici $\Gamma = \bar{\Gamma}_1 \cup \bar{\Gamma}_2$

²Neumannova okrajová podmínka je speciálním případem pro $\alpha = 0$.

kde $p = p(x, y)$, $q = q(x, y)$, $g = g(x, y)$, $f = f(x, y)$, $\alpha = \alpha(x, y)$, $\beta = \beta(x, y)$ jsou dané funkce.

Řešení

V praktických úlohách většinou nelze vystačit s tzv. klasickým řešením $u \in C^2(\overline{\Omega})$ okrajové úlohy (2.1), (2.2), (2.3), protože požadavky kladené na data úlohy (funkce $p, q, g, f, \alpha, \beta$) pro existenci tohoto klasického řešení jsou příliš přísné. Lze hledat tzv. slabé řešení, jehož existence je v praktických úlohách většinou zaručena. Dále budou uplatněny následující předpoklady

- Ω je mnohoúhelník,
- $p \geq p_0 > 0$, $q \geq 0$, g jsou po částech spojité v Ω ,
- f je spojitá na Γ_1 ,
- $\alpha \geq 0$, β jsou po částech spojité na Γ_2 ,
- jestliže $\Gamma = \Gamma_2$ pak je buď $q \geq q_0 > 0$ na části Ω , nebo $\alpha \geq \alpha_0 > 0$ na části Γ_2 ,

pak existuje jediné slabé řešení úlohy (2.1), (2.2), (2.3).

Jestliže platí $\Gamma = \Gamma_2$, $q = \alpha = 0$ a platí $\int_{\Omega} g dx + \int_{\Gamma} \beta dS = 0$, pak existuje nekonečně mnoho slabých řešení úlohy (2.1), (2.2), (2.3), která se liší o konstantu.

Slabá formulace úlohy

Rovnice (2.1) je vynásobena testovací funkcí $v \in C^1(\Omega)$, která nabývá nulové hodnoty na Γ_1 . Poté je rovnice integrována přes Ω

$$-\int_{\Omega} [\nabla \cdot (p \nabla u) v + q u v] dx = \int_{\Omega} g v dx.$$

První člen integrandu na levé straně rovnice lze upravit pomocí Gaussovy-Ostrogradského věty

$$-\int_{\partial \Omega} p \nabla u \cdot \mathbf{n} v dS + \int_{\Omega} p \nabla u \cdot \nabla v dx + \int_{\Omega} q u v dx = \int_{\Omega} g v dx$$

Dále lze využít toho, že $v = 0$ na Γ_1 , že na Γ_2 je předepsána Newtonova okrajová podmínka a že $\frac{\partial u}{\partial n} = \nabla u \cdot \mathbf{n}$, tedy

$$\int_{\Omega} p \nabla u \cdot \nabla v dx + \int_{\Omega} q u v dx + \int_{\Gamma_2} \alpha u v dS = \int_{\Omega} g v dx + \int_{\Gamma_2} \beta v dS.$$

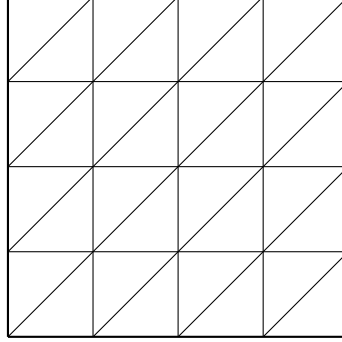
Levá část této rovnice výše je označena $a(u, v)$ a pravá $L(v)$.

Úloha najít $u \in W = \{v \in X \mid v = f \text{ na } \Gamma_1\}$, kde $X = PC^1(\overline{\Omega})$, splňující $a(u, v) = L(v)$, $\forall v \in V = \{v \in X \mid v = 0 \text{ na } \Gamma_1\}$ se nazývá slabou formulací úlohy (2.1), (2.2), (2.3) a u se nazývá slabým řešením. $PC^1(\overline{\Omega})$ označuje prostor funkcí, které jsou v $\overline{\Omega}$ spojité a jejichž první derivace jsou v Ω po částech spojité.

Triangulace oblasti Ω

Uzávěr mnohoúhelníku Ω je vyjádřen jako sjednocení konečného počtu uzavřených trojúhelníků T_e . Každé dva z těchto trojúhelníků jsou buď disjunkt, nebo mají společný vrchol, nebo společnou stranu. Množina $\mathcal{T} = \{T_1, T_2, \dots, T_{N_T}\}$ těchto trojúhelníků se nazývá triangulace oblasti Ω .

Vrcholy trojúhelníků se nazývají uzly a jsou označeny P_1, P_2, \dots, P_M . Dále se předpokládá, že společné body $\bar{\Gamma}_1, \bar{\Gamma}_2$ jsou uzly triangulace. Množina stran $T \in \mathcal{T}$, jejichž sjednocení dá $\bar{\Gamma}_2$ je označena $\mathcal{S} = \{S_1, S_2, \dots, S_{N_S}\}$. Příklad triangulace mnohoúhelníku je ukázán na obrázku 2.1.



Obrázek 2.1: Příklad triangulace čtverce

Diskrétní slabá formulace

Funkce spojitá na $\bar{\Omega}$ a lineární na každém trojúhelníku $T \in \mathcal{T}$ se nazývá spojitá po částech lineární funkce. Prostor všech spojitých po částech lineárních funkcí se označuje X_h . Speciálním typem funkcí v prostoru X_h jsou bázové funkce w_i definované jako $w_i(x_j, y_j) = \delta_{ij}$, kde $(x_j, y_j) = P_j$ jsou souřadnice uzlu triangulace a δ_{ij} je Kroneckerovo delta.

Libovolnou funkci $v \in X_h$ lze vyjádřit jako

$$v(x, y) = \sum_{i=1}^M v(x_i, y_i) w_i(x, y).$$

Analogicky jako ve spojité slabé formulaci je definován prostor testovacích funkcí $V_h = \{v \in X_h \mid v(x_i, y_i) = 0 \forall P_i \in \bar{\Gamma}_1\}$ a množina přípustných řešení $W_h = \{v \in X_h \mid v(x_i, y_i) = f(x_i, y_i) \forall P_i \in \bar{\Gamma}_1\}$. Úloha najít $U \in W_h$ splňující $a_h(U, v) = L_h(v) \forall v \in V_h$ se nazývá diskrétní slabá formulace úlohy (2.1), (2.2), (2.3), kde

$$a_h(U, v) = \sum_{T_e \in \mathcal{T}} Q^{T_e}(p \nabla U \cdot \nabla v) + Q^{T_e}(q U v) + \sum_{S_e \in \mathcal{S}} Q^{S_e}(\alpha U v), \quad (2.4)$$

$$L_h(v) = \sum_{T_e \in \mathcal{T}} Q^{T_e}(g v) + \sum_{S_e \in \mathcal{S}} Q^{S_e}(\beta v), \quad (2.5)$$

kde $Q^{T_e}(\varphi)$ je kvadrurní formule aproximující $\int_{T_e} \varphi \, dx$ a $Q^{S_e}(\varphi)$ je kvadrurní formule aproximující $\int_{S_e} \varphi \, dS$.

Vhodné kvadrurní formule

Podle [4] jsou doporučeny tyto kvadrurní formule

- Pro numerický výpočet integrálů obsahující integrand $p\nabla U \cdot \nabla v$ lze použít formuli

$$Q^T(\varphi) = |T|\varphi(P_0), \quad (2.6)$$

kde $|T|$ je plocha trojúhelníka T a $P_0 = (x_0, y_0)$ jsou souřadnice těžiště trojúhelníka T .

- Pro výpočet integrálů obsahující integrandy qUv, gv lze použít formuli

$$Q^T(\varphi) = \frac{1}{3}|T|[\varphi(P_1) + \varphi(P_2) + \varphi(P_3)], \quad (2.7)$$

kde P_1, P_2, P_3 jsou vrcholy trojúhelníka T .

- Pro výpočet integrálů obsahující $\alpha Uv, \beta v$ lze použít formuli

$$Q^S(\varphi) = \frac{1}{2}|S|[\varphi(P_1) + \varphi(P_2)], \quad (2.8)$$

kde $|S|$ je délka strany S a P_1, P_2 jsou koncové body strany S .

Další používané kvadraturní formule je možné najít v [5].

Soustava rovnic

Bez újmy na obecnosti lze předpokládat, že uzly ležící uvnitř Ω , nebo na hranici Γ_2 mají indexy $1, 2, \dots, N$ a že uzly ležící na hranici Γ_1 mají indexy $N + 1, N + 2, \dots, M$, pak lze funkce U, v vyjádřit ve tvaru

$$U(x, y) = \sum_{j=1}^N \Delta_j w_j(x, y) + \sum_{j=N+1}^M f_j w_j(x, y), \quad v(x, y) = \sum_{i=1}^N \theta_i w_i(x, y),$$

kde $\Delta_j = U(P_j)$, $f_j = f(P_j)$, $\theta_i = v(P_i)$. Dosazením do slabé diskretní formulace a s využitím linearitu a_h, v_h lze dostat soustavu rovnic

$$\boldsymbol{\theta}^\top (\mathbf{K}\boldsymbol{\Delta} - \mathbf{F}) = 0, \quad (2.9)$$

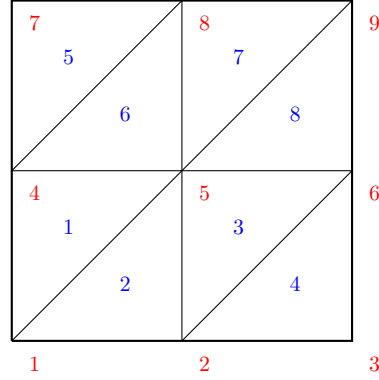
kde $\mathbf{K} = \{k_{ij}\}_{i,j=1}^N$, $k_{ij} = a_h(w_j, w_i)$, $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_N)^\top$, $\boldsymbol{\Delta} = (\Delta_1, \Delta_2, \dots, \Delta_N)^\top$ a $\mathbf{F} = (F_1, F_2, \dots, F_N)^\top$, $F_i = L_h(w_i) - \sum_{j=N+1}^M f_j a_h(w_j, w_i)$. Jelikož funkce v je libovolná, tak i vektor $\boldsymbol{\theta}$ je libovolný a tedy musí platit

$$\mathbf{K}\boldsymbol{\Delta} = \mathbf{F}.$$

Řešením této soustavy lineárních rovnic jsou hodnoty U v uzlech triangulace \mathcal{T} . Matice \mathbf{K} je pozitivně definitní (z toho plyne, že je regulární) a nazývá se globální matice tuhosti. Vektor \mathbf{F} se nazývá globální vektor zatížení. Globální matice tuhosti a globální vektor zatížení se sestavuje z lokálních matic tuhosti a lokálních vektorů zatížení, právě toto sestavení vyžaduje největší úsilí při numerickém řešení úlohy eliptického typu.

Sestavení matice \mathbf{K} a vektoru \mathbf{F}

Trojúhelníky triangulace \mathcal{T} jsou zapsány pomocí čísel uzlů do tabulky, která má N_T řádků a tři sloupce. Triangulace na obrázku 2.2, kde jsou čísla uzlů znázorněna červeně a čísla trojúhelníků modře, pak dává tabulku 2.1



Obrázek 2.2: Příklad očíslování trojúhelníků a uzlů

	1	2	3
1	1	4	5
2	1	2	5
3	2	5	6
4	2	3	6
5	4	7	8
6	4	5	8
7	5	8	9
8	5	6	9

Tabulka 2.1: Tabulka pro triangulaci na obrázku 2.2

Nechť $T_e \in \mathcal{T}$ označuje jeden konkrétní trojúhelník triangulace. Jeho vrcholy jsou označeny $P_1^e = (x_1, y_1)$, $P_2^e = (x_2, y_2)$, $P_3^e = (x_3, y_3)$. Globální index vrcholů trojúhelníku P_1 , P_2 , P_3 jsou v tabulce 2.1 na řádce e , ve sloupci 1, 2, 3 po řadě. Restrikce slabého řešení U a testovací funkce v na trojúhelník T_e jsou po řadě označeny takto

$$U^e(x, y) = \Delta_1^e w_1^e(x, y) + \Delta_2^e w_2^e(x, y) + \Delta_3^e w_3^e(x, y),$$

$$v^e(x, y) = \theta_1^e w_1^e(x, y) + \theta_2^e w_2^e(x, y) + \theta_3^e w_3^e(x, y),$$

kde $\Delta_r^e = U(P_r^e)$, $\theta_r^e = v(P_r^e)$ a $w_r^e = a_r^e x + b_r^e y + c_r^e$, $r = 1, 2, 3$ je bázová funkce příslušná uzlu P_r^e . Snadno lze odvodit, že platí $\nabla U^e = \mathbf{B}^e \mathbf{\Delta}^e$, $\nabla v^e = \mathbf{B}^e \mathbf{\theta}^e$, kde

$$\mathbf{B}^e = \begin{pmatrix} a_1^e & a_2^e & a_3^e \\ b_1^e & b_2^e & b_3^e \end{pmatrix}, \quad \mathbf{\Delta}^e = (\Delta_1^e, \Delta_2^e, \Delta_3^e)^\top, \quad \mathbf{\theta}^e = (\theta_1^e, \theta_2^e, \theta_3^e)^\top.$$

Pro první člen z (2.4), použitím kvadraturní formule (2.6), lze dostat $Q^{T_e}(p \nabla U^e \cdot \nabla v^e) = |T_e| p(P_0^e) (\mathbf{B}^e \mathbf{\theta}^e)^\top (\mathbf{B}^e \mathbf{\Delta}^e) = (\mathbf{\theta}^e)^\top \mathbf{K}^{e_1} \mathbf{\Delta}^e$. Podle [5] a pomocí vlastností bázových funkcí

$w_i^T(x_j, y_j) = \delta_{ij}$ lze vypočítat, že matice \mathbf{K}^{e1} je tvaru

$$\mathbf{K}^{e1} = \frac{p(P_0^e)}{2|d^e|} \begin{pmatrix} -r^e - s^e & r^e & s^e \\ r^e & -r^e - t^e & t^e \\ s^e & t^e & -s^e - t^e \end{pmatrix},$$

kde

$$\begin{aligned} r^e &= (y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3), \\ s^e &= (y_2 - y_3)(y_1 - y_2) + (x_3 - x_2)(x_2 - x_1), \\ t^e &= (y_3 - y_1)(y_1 - y_2) + (x_1 - x_3)(x_2 - x_1), \\ d^e &= (y_3 - y_1)(x_2 - x_1) - (x_1 - x_3)(y_1 - y_2). \end{aligned}$$

Pro další členy (2.4) se postupuje analogicky, použitím formule (2.7) se dostane $Q_e^T(qUv) = (\boldsymbol{\theta}^e)^\top \mathbf{K}^{e2} \boldsymbol{\Delta}^e$, kde

$$\mathbf{K}^{e2} = \frac{|T_e|}{3} \begin{pmatrix} q(P_1^e) & 0 & 0 \\ 0 & q(P_2^e) & 0 \\ 0 & 0 & q(P_3^e) \end{pmatrix}.$$

Lokální matici tuhosti lze dostat jako $\mathbf{K}^e = \mathbf{K}^{e1} + \mathbf{K}^{e2}$. Nakonec pro lokální vektor zatížení v (2.5), opět užitím kvadraturní formule (2.7), lze odvodit $Q^T(fv) = (\boldsymbol{\theta}^e)^\top \mathbf{F}^e$, kde

$$\mathbf{F}^e = \frac{|T_e|}{3} \begin{pmatrix} g(P_1^e) \\ g(P_2^e) \\ g(P_3^e) \end{pmatrix}.$$

Dále je třeba sestavit elementární matici a vektor na stranách trojúhelníků, které leží na hranici Γ_2 . Odvození vztahů se provádí analogicky jako na prvku, je sestavena tabulka, která má N_S řádků a dva sloupce a do ní jsou zaznamenány indexy krajních bodů stran $S \in \mathcal{S}$. Nechtě $S_e \in \mathcal{S}$ označuje konkrétní stranu. Koncové body této strany jsou označeny $P_1^S = (x_1^e, y_1^e)$, $P_2^S = (x_2^e, y_2^e)$. Globální indexy těchto bodů jsou v tabulce na e -tém řádku. Restrikce řešení U a funkce v na stranu S jsou označeny takto

$$\begin{aligned} U^e(x, y) &= \Delta_1^e w_1^e(x, y) + \Delta_2^e w_2^e(x, y), \\ v^e(x, y) &= \theta_1^e w_1^e(x, y) + \theta_2^e w_2^e(x, y), \end{aligned}$$

kde $\Delta_r^e = U(P_r^e)$, $\theta_r^e = v(P_r^e)$ a w_r^e je bázová funkce příslušná uzlu P_r^e , $r = 1, 2$. Užitím formule (2.8) lze odvodit $Q^{S_e}(\alpha Uv) = (\boldsymbol{\theta}^{S_e})^\top \mathbf{K}^{S_e} \boldsymbol{\Delta}^{S_e}$, kde

$$\mathbf{K}^{S_e} = \frac{|S_e|}{2} \begin{pmatrix} \alpha(P_1^e) & 0 \\ 0 & \alpha(P_2^e) \end{pmatrix}, \quad \boldsymbol{\Delta}^{S_e} = \begin{pmatrix} \Delta_1^e \\ \Delta_2^e \end{pmatrix}, \quad \boldsymbol{\theta}^{S_e} = \begin{pmatrix} \theta_1^e \\ \theta_2^e \end{pmatrix}.$$

Nakonec, užitím (2.8), lze dostat $Q^{S_e}(\beta v) = (\boldsymbol{\theta}^{S_e})^\top \mathbf{F}^{S_e}$, kde

$$\mathbf{F}^{S_e} = \frac{|S_e|}{2} \begin{pmatrix} \beta(P_1^e) \\ \beta(P_2^e) \end{pmatrix}.$$

Všechny výše uvedené lokální vztahy jsou nyní dosazeny do rovnice (2.9), čímž je odvozeno

$$0 = a_h(U, v) - L_h(v) = \boldsymbol{\theta}^\top (\mathbf{K} \boldsymbol{\Delta} - \mathbf{F}) = \sum_{T_e \in \mathcal{T}} (\boldsymbol{\theta}^e)^\top (\mathbf{K}^e \boldsymbol{\Delta}^e - \mathbf{F}^e) + \sum_{S_e \in \mathcal{S}} (\boldsymbol{\theta}^{S_e})^\top (\mathbf{K}^{S_e} \boldsymbol{\Delta}^{S_e} - \mathbf{F}^{S_e}).$$

Z této rovnice lze odvodit algoritmus pro sestavování globální matice \mathbf{K} a vektoru \mathbf{F} . Algoritmus, který byl implementován autorem práce, je možné nalézt v [5].

2.2 Parabolické úlohy

Podobně jako v případě eliptické úlohy je nejprve formulována parabolická úloha. Dále je tato úloha formulována ve slabé formě. Poté je provedena prostorová diskretizace podobně, jako u úloh eliptického typu, tím vznikne počáteční problém pro soustavu obyčejných diferenciálních rovnic. Nakonec je tato soustava vyřešena θ -metodou. Podrobnosti lze nalézt v [4] a v [5].

Formulace úlohy

Úkolem bude aproximovat funkci $u = u(x, y, t)$ splňující rovnici

$$c \frac{\partial u}{\partial t} - \nabla \cdot (p \nabla u) + qu = g, \quad \mathbf{x} \in \Omega, \quad t \in (0, T), \quad (2.10)$$

okrajové podmínky

$$\begin{aligned} u &= f, \quad \mathbf{x} \in \Gamma_1, \quad t \in (0, T) \\ -p \frac{\partial u}{\partial n} &= \alpha u - \beta, \quad \mathbf{x} \in \Gamma_2, \quad t \in (0, T), \end{aligned} \quad (2.11)$$

a počáteční podmínku

$$u(x, y, 0) = \varphi(x, y), \quad \mathbf{x} \in \Omega, \quad (2.12)$$

kde $c = c(x, y, t)$, $p = p(x, y, t)$, $q = q(x, y, t)$, $g = g(x, y, t)$, $f = f(x, y, t)$, $\alpha = \alpha(x, y, t)$, $\beta = \beta(x, y, t)$, $\varphi = \varphi(x, y)$ jsou dané funkce.

Slabá formulace

Slabá formulace počátečně-okrajového problému (2.10), (2.11), (2.12) je odvozena analogicky jako v případě eliptické úlohy. Rovnice (2.10) je opět vynásobena testovací funkcí $v \in V$ a integrována přes Ω , oproti eliptické úloze se zde nachází navíc člen $(cu_t, v) = \int_{\Omega} cu_tv \, d\mathbf{x}$, tedy

$$(cu_t, v) + a(u, v) = L(v), \quad \forall v \in V, \quad t \in (0, T),$$

kde $a(u, v)$, $b(v)$ jsou stejné jako v úloze eliptického typu.

Diskretizace v prostoru

Provádí se stejně jako v úloze eliptického typu, nicméně hodnoty přibližného řešení v uzlech triangulace jsou nyní funkcemi času $U_j = U_j(t) = U(x_j, y_j, t)$. Na aproximaci členů $a(u, v)$, $L(v)$ se ovšem nic nemění. Zbývá tedy aproximovat člen (cu_t, v) , ten je nahrazen výrazem

$$(cU_t, v)_h = \sum_{T_e \in \mathcal{T}} Q^{T_e}(cU_tv).$$

Jelikož platí $U_t = \sum_{j=1}^N \dot{U}_j w_j(x, y)$ ³, tak, užitím kvadraturní formule (2.7), lze dostat $Q^{T_e}(cU_tv) = (\boldsymbol{\theta}^e)^\top \mathbf{C}^e \boldsymbol{\Delta}^e$, kde

$$\mathbf{C}^e = \frac{|T_e|}{3} \begin{pmatrix} c(P_1^e) & 0 & 0 \\ 0 & c(P_2^e) & 0 \\ 0 & 0 & c(P_3^e) \end{pmatrix},$$

³Tečka nad \dot{U}_j značí derivaci podle času.

a kde $\dot{\Delta}^T = (\dot{\Delta}_1^T, \dot{\Delta}_2^T, \dot{\Delta}_3^T)^\top$. Globální matice \mathbf{C} je stejně jako matice lokální \mathbf{C}^e diagonální. Po složení lokálních matic do matice globální lze dostat počáteční problém pro soustavu obyčejných diferenciálních rovnic

$$\mathbf{C}\dot{\Delta} + \mathbf{K}\Delta = \mathbf{F}, \quad \Delta(0) = \varphi, \quad (2.13)$$

kde $\varphi = (\varphi(x_1, y_1), \varphi(x_2, y_2), \dots, \varphi(x_M, y_M))^\top$ je počáteční podmínka vyhodnocená v uzlech triangulace.

θ -metoda

Jednou z možností jak řešit soustavu (2.13) je θ -metoda, která bude nyní odvozena. Interval $(0, T)$ je rozdělen na body $0 = t_0 < t_1 < \dots < t_Q = T$. Nechť $\tau_n = t_{n+1} - t_n$ označuje délku časového kroku, kde $n = 0, 1, \dots, Q - 1$. Poté je zvoleno pevné číslo $\theta \in \langle 0, 1 \rangle$ a nechť $t_{n+\theta} = t_n + \tau_n\theta$. Dále nechť Δ^n označuje aproximaci $\Delta(t_n)$, pak $\Delta^0 = \varphi$. Rovnice (2.13) je napsána v čase $t_{n+\theta}$

$$\mathbf{C}\dot{\Delta}^{n+\theta} + \mathbf{K}\Delta^{n+\theta} = \mathbf{F}^{n+\theta}, \quad (2.14)$$

kde $\mathbf{F}^{n+\theta} = \mathbf{F}(t_{n+\theta})$. Jestliže se navíc využije předpoklad, že $\Delta(t)$ je na $\langle t_n, t_{n+1} \rangle$ lineární, pak, jak se lze přesvědčit z obrázku 2.3, lze odvodit

$$\Delta(t) = \Delta^n + \frac{t - t_n}{\tau_n}(\Delta^{n+1} - \Delta^n).$$

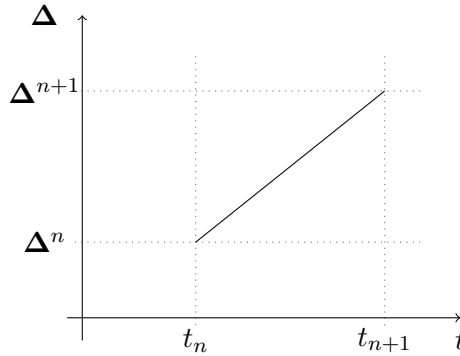
Z tohoto vztahu okamžitě plynou vztahy

$$\begin{aligned} \dot{\Delta}^{n+\theta} &= \frac{\Delta^{n+1} - \Delta^n}{\tau_n}, \\ \Delta^{n+\theta} &= (1 - \theta)\Delta^n + \theta\Delta^{n+1}. \end{aligned}$$

Oba výše uvedené vztahy jsou dosazeny do (2.14), čímž je odvozena θ -metoda

$$[\mathbf{C} + \tau_n\theta\mathbf{K}]\Delta^{n+1} = [\mathbf{C} - \tau_n(1 - \theta)\mathbf{K}]\Delta^n + \tau_n\mathbf{F}^{n+\theta}.$$

Řešením této soustavy lineárních rovnic je aproximace řešení v uzlech triangulace v čase t_{n+1} .



Obrázek 2.3: Lineární průběh Δ na intervalu $\langle t_n, t_{n+1} \rangle$

0-metoda je explicitní Eulerovo schéma, 1-metoda je implicitní Eulerova schéma a pro $\theta = \frac{1}{2}$ se jedná o tzv. Crankovu-Nicolsonovu metodu. Crankova-Nicolsonova metoda je

řádu přesnosti dva, z tohoto důvodu se používá nejčastěji. Pokud ovšem nejsou zaručeny tzv. podmínky souladu (počátečních a okrajových podmínek), tak je vhodné prvních několik kroků počítat implicitní Eulerovou metodou, která je stabilnější a poté se přejít na Crankovu-Nicolsonovu metodu (viz [5]). Pro $\theta \in \langle \frac{1}{2}, 1 \rangle$ je toto schéma stabilní (bez vlivu velikosti kroku).

2.3 Nelineární úlohy

Jestliže některá z funkcí p, q, g, α, β v úloze (2.1),(2.2),(2.3), případně také c v rovnici (2.10) závisí na u , pak je úloha nelineární. Jak bylo řečeno v první kapitole, tak například součinitel tepelné vodivosti u většiny materiálů závisí na teplotě, také úlohy vedení tepla se změnou fáze lze formulovat jako úlohy s nelineární tepelnou kapacitou, což ukazuje důležitost nelineárních úloh.

Stacionární úlohy

Diskretizace metodou konečných prvků probíhá formálně naprosto stejně jako v lineárním případě, podstatný rozdíl je v tom, že výsledná soustava rovnic je nelineární, tedy

$$\mathbf{K}(\Delta)\Delta = \mathbf{F}(\Delta).$$

Tuto soustavu je možné řešit nějakou vhodnou iterační metodou, např. metodou prosté iterace nebo Newtonovou metodou. V obou metodách je podstatná počáteční aproximace řešení $\Delta^{(0)}$. Nejjednodušší je metoda prosté iterace, která je zapsána jako

$$\mathbf{K}(\Delta^{(k)})\Delta^{(k+1)} = \mathbf{F}(\Delta^{(k)}).$$

Z této rovnice je vypočteno $\Delta^{(k+1)}$ a to je poté použito pro získání nové iterace. Výpočet pokračuje, dokud vektor reziduí $\mathbf{R} = \mathbf{K}(\Delta^{(k)})\Delta^{(k)} - \mathbf{F}(\Delta^{(k)})$ není dostatečně malý, nebo se dvě po sobě jdoucí iterace $\Delta^{(k)}, \Delta^{(k+1)}$ neliší jen málo. Metoda prosté iterace konverguje k řešení spíše výjimečně. Častěji se používá Newtonova metoda, která je zapsána jako

$$\begin{aligned} \mathbf{H}(\Delta^{(k)})\mathbf{Y}^{(k)} &= -\mathbf{R}(\Delta^{(k)}), \\ \Delta^{(k+1)} &= \Delta^{(k)} + \mathbf{Y}^{(k)}, \end{aligned}$$

kde \mathbf{H} je Jacobiho matice vektoru reziduí $\mathbf{H} = (\nabla\mathbf{R})^\top$. Postup sestavení této matice z matic lokálních lze nalézt v [5].

Nestacionární úlohy

Také diskretizace v případě nestacionární úlohy probíhá formálně stejně jako v lineárním případě, ovšem výsledná soustava obyčejných diferenciálních rovnic je nelineární

$$\mathbf{C}(\Delta)\dot{\Delta} + \mathbf{K}(\Delta)\Delta = \mathbf{F}(t, \Delta).$$

Podle [4] lze pro vyřešení tohoto počátečního problému v Matlabu použít např. funkci ode23t, což je implementace lichoběžníkové metody. Podle [5] lze použít také θ -metodu. Její aplikací je možné dostat nelineární soustavu rovnic tvaru

$$[\mathbf{C}^{n+\theta} + \tau_n\theta\mathbf{K}^{n+\theta}]\Delta^{n+1} = [\mathbf{C}^{n+\theta} - \tau_n(1-\theta)\mathbf{K}^{n+\theta}]\Delta^n + \tau_n\mathbf{F}^{n+\theta},$$

kde $\mathbf{C}^{n+\theta} = \mathbf{C}(t_{n+\theta}, \Delta^{n+\theta})$, $\mathbf{K}^{n+\theta} = \mathbf{K}(t_{n+\theta}, \Delta^{n+\theta})$, $\mathbf{F}^{n+\theta} = \mathbf{F}(t_{n+\theta}, \Delta^{n+\theta})$, kde $\Delta^{n+\theta} = \Delta^n + \theta(\Delta^{n+1} - \Delta^n)$. Pro získání řešení v čase t_{n+1} je tedy nutno vyřešit nelineární soustavu rovnic. Tuto soustavu rovnic se řeší vhodnou numerickou metodou.

Kapitola 3

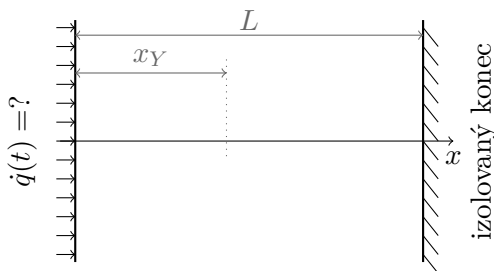
Inverzní úlohy přenosu tepla

Jestliže jsou fyzikální vlastnosti látky, počáteční, okrajové podmínky a tvar tělesa známé, pak vyřešením přímé úlohy vedení tepla, je nalezeno rozložení teploty v tělese a v čase. Úloha, ve které není některý z těchto údajů znám, se nazývá inverzní úloha vedení tepla. Řešení inverzních úloh vedení tepla je obvykle obtížnější než řešení přímé úlohy vedení tepla, z matematického hlediska se jedná o tzv. špatně podmíněné úlohy.

Jedna z prvních publikací, která se zabývala inverzní úlohou vedení tepla, byla vydána v roce 1960 G. Stolzem, který řešil problém vedení tepla při kalení součástí jednoduchých tvarů. Stolzův přístup měl nevýhodu v nestabilitě při malém časovém kroku. Významný impulz pro rozvoj metod řešení inverzních úloh vedení tepla byl začátek kosmického programu v roce 1956. Jedním z výzkumníků byl i J. V. Beck. Beckova metoda umožňovala výpočet při menších časových krocích než Stolzova metoda. Aplikace v kosmickém programu byla při výpočtu hustoty tepelného toku v hlavicích střel, vesmírných sond a podobných zařízení. Další oblastí, ve které se vyskytla potřeba tyto úlohy řešit, je testování součástí jaderných reaktorů [2].

3.1 Formulace inverzní úlohy vedení tepla

Je uvažováno vedení tepla v tělese tvaru obdélníku (vedení probíhá pouze ve směru osy x viz obrázek 3.1), šířka obdélníku je L , jeho materiál je homogenní, izotropní a v tělese se negeneruje žádné teplo. Počáteční teplota stěny je konstantní a je označena T_0 . Na levém konci je předepsána hustota tepelného toku jako funkce času $\dot{q} = \dot{q}(t)$. Pravý konec je tepelně izolovaný. Teplotu měří teplotní senzor umístěný ve vzdálenosti x_Y od levého okraje stěny, naměřené hodnoty teploty v časech t_i jsou označeny Y_i , $i = 1, \dots, N$. Úlohou je určit neznámou funkci \dot{q} .



Obrázek 3.1: Inverzní úloha

Nyní bude zapsán matematický model této inverzní úlohy. Vývoj teploty $T = T(x, t)$ je popsán jednorozměrnou verzí rovnice vedení tepla

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right), \quad x \in \langle 0, L \rangle, \quad t \in (0, T).$$

V počátečním čase je předepsána počáteční teplota T_0

$$T(x, 0) = T_0, \quad x \in \langle 0, L \rangle.$$

Pravý konec je tepelně izolován, tedy je předepsán nulový tepelný tok

$$-\lambda \frac{\partial T(L, t)}{\partial x} = 0, \quad t \in (0, T).$$

Požadavkem na funkci \dot{q} je, aby se teplota v místě, kde je měřena teplota, shodovala s naměřenými údaji

$$T(x_Y, t_i) = Y_i, \quad i = 1, \dots, N.$$

Dále také musí platit, že hustota tepelného toku \dot{q} v časech měření teploty t_i je rovna

$$\dot{q}(t_i) = -\lambda \frac{\partial T(0, t_i)}{\partial x}, \quad i = 1, \dots, N.$$

Výše popsaná úloha je složitější než přímá úloha vedení tepla, namísto okrajové podmínky je zadána teplota, avšak pouze v diskrétních bodech (x_Y, t_i) . Dalším problémem je, že měření teploty může obsahovat nějakou chybu. Jestliže některá z vlastností ρ , c a λ v rovnici vedení tepla závisí na teplotě, pak se inverzní úloha vedení tepla nazývá nelineární [2].

s výše formulovanou inverzní úlohou je těsně spjata inverzní úloha, ve které je namísto hustoty tepelného toku určován součinitel přestupu tepla. Mezi další inverzní problémy se řadí tzv. Cauchyho problém pro Laplaceovu rovnici, ve které je z několika naměřených hodnot rekonstruována teplota v celém tělese. Mezi inverzní úlohy lze zařadit také tzv. zpětné vedení tepla, ve kterém je cílem určit počáteční teplotu T_0 se znalostí podmínek okrajových a měření teploty v několika bodech. Ve dvourozměrném případě lze také, kromě závislosti tepelného toku na čase, předpokládat jeho závislost v prostoru (tj. $\dot{q} = \dot{q}(t, x, y)$). Další kategorií inverzních úloh je také určování fyzikálních vlastností materiálu (tj. ρ , c , λ) z vývoje systému v čase.

Měření teploty

Měření jakékoliv fyzikální veličiny je vždy zatíženo nějakou nepřesností, chybou. Cílem tohoto odstavce je tuto chybu statisticky popsat. Předpokládá se, že chyba má náhodný charakter a všechny ostatní „nenáhodné“ vlivy jsou odstraněny (např. vhodnou kalibrační senzory). Pro popis chyby měření teploty lze použít těchto osm předpokladů (viz [2]):

1. Chyby jsou aditivní, tedy $Y_i = T_i + \varepsilon_i$, kde Y_i je teplota naměřená v čase t_i , T_i je „skutečná“ teplota, nezatížená chybou, v čase t_i a ε_i je náhodná chyba měření v čase t_i .
2. Chyby mají nulovou střední hodnotu: $E(\varepsilon_i) = 0$.
3. Měření teploty má konstantní rozptyl: $D(Y_i) = \sigma^2$. Tato podmínka říká že rozptyl měření teploty nezávisí na čase.

4. Chyby jsou nekorelované: $\rho(\varepsilon_i, \varepsilon_j) = 0, i \neq j$.
5. Chyby měření mají normální rozdělení s hustotou pravděpodobnosti

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}.$$

6. Parametr σ^2 je známý.
7. Měření času, polohy teplotního senzoru, rozměrů tělesa, fyzikálních vlastností látky tělesa jsou známá a chyba jejich měření je vůči chybě měření teploty zanedbatelná.
8. Není k dispozici žádná apriorní informace o hustotě tepelného toku. Pokud by bylo známo, že např. hustota tepelného toku má periodický průběh, lze tuto informaci použít pro zlepšení odhadu funkce \dot{q} .

Metody řešení

Metody řešení inverzních úloh mohou být klasifikovány podle několika kritérií [2]. Jedno z nich je, zda metoda je schopna řešit lineární i nelineární problémy. Obě dále popsané metody (sekvenční algoritmus a neuronové sítě) lze použít jak na lineární, tak i na nelineární inverzní úlohy. Dalším kritériem může být to, jakou metodou je řešena přímá úloha vedení tepla (Duhamelův princip, metoda konečných diferencí, objemů, prvků, ...). Počet použitých měření teploty při řešení, může také sloužit k rozdělení metod. Používá-li metoda pouze teploty naměřené do současného časového kroku, jedná se o Stolzovu metodu. Její nevýhoda je ve velké citlivosti na chyby v měření. Použije-li metoda všechny naměřené teploty do současného časového kroku a k tomu několik dalších teplot, jedná se o sekvenční algoritmus. Sekvenční algoritmus je méně citlivý na chyby než Stolzův algoritmus, navíc dovoluje zmenšit časový krok. Celodoménový algoritmus má podobné vlastnosti, nicméně není tak výpočetně efektivní. K řešení inverzních úloh lze použít také metody založené na umělých neuronových sítích [12].

3.2 Metoda odhadu funkce

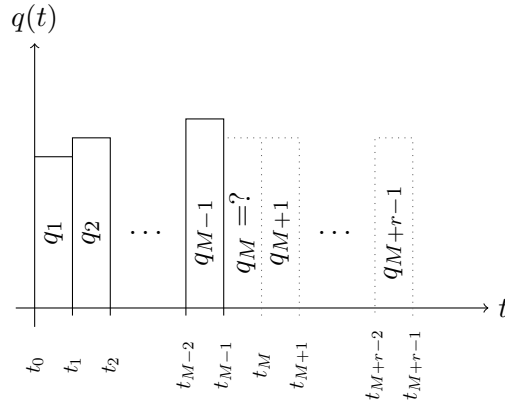
Jednou z možností jak řešit inverzní úlohu vedení tepla je, podle [2], předpokládat, že závislost hustoty tepelného toku na čase je určitého typu např. po částech konstantní, nebo po částech lineární. K odhadu parametrů této funkce je pak možno přistoupit buď celodoménově, nebo sekvenčně. V celodoménové metodě jsou odhadnuty všechny parametry funkce najednou, zatímco při sekvenční metodě jsou parametry odhadovány postupně, jeden za druhým. Sekvenční metoda je výpočetně efektivnější než metoda celodoménová.

Sekvenční metoda

Sekvenční metoda je velmi známou metodou řešení inverzních úloh vedení tepla, často se jí také říká Beckova sekvenční metoda. Princip metody bude vysvětlen na inverzní úloze ze sekce 3.1. Hustota tepelného toku $\dot{q} = \dot{q}(t)$ je v časových intervalech $(t_0, t_1), (t_1, t_2), \dots, (t_{M-2}, t_{M-1})$ odhadnuta konstantami q_1, q_2, \dots, q_{M-1} . Cílem metody je určit konstantu q_M (viz obrázek 3.2) odhadující funkci \dot{q} na intervalu (t_{M-1}, t_M) . Dočasně je použit předpoklad

$$q_M = q_{M+1} = \dots = q_{M+r-1},$$

který metodě dodává numerickou stabilitu.



Obrázek 3.2: M -tý krok sekvenčního algoritmu

Konstanta q_M je určena minimalizací součtu čtverců odchylek

$$S(q_M) = \sum_{i=1}^r (Y_{M+i-1} - T_{M+i-1})^2,$$

kde $T_M, T_{M+1}, \dots, T_{M+r-1}$ jsou teploty vypočtené řešením přímé úlohy vedení tepla v časech $t_M, t_{M+1}, \dots, t_{M+r-1}$ a v místě teplotního senzoru (závisí na q_M) a kde $Y_M, Y_{M+1}, \dots, Y_{M+r-1}$ jsou teploty naměřené teplotním senzorem v odpovídajících časech. Poté je M zvýšen o jedna a výpočet opakován, dokud je k dispozici dostatek budoucích kroků k výpočtu, tj. do $N - r$. Postup výpočtu je shrnut pomocí pseudokódu.

Algoritmus 1: Sekvenční metoda

- 1: zvol vhodné $r \in \mathbb{N}$, vynuluj \mathbf{q}
 - 2: **for** $M = 1$ to $N - r$ **do**
 - 3: předpokládej: $q_M = q_{M+1} = \dots = q_{M+r-1} = konst.$
 - 4: výpočet q_M : $q_M = \arg \min S(q_M) = \arg \min \sum_{j=M}^{M+r-1} (Y_j - T(x_Y, t_j))^2$
 - 5: ulož výsledek $\mathbf{q}[M] = q_M, M = M + 1$
 - 6: **end for**
-

Výpočet hodnoty účelové funkce S je poměrně výpočetně náročný, znamená totiž řešit přímou úlohu vedení tepla pro r časových kroků. Navíc obecně není známa derivace účelové funkce S podle q_M . Z těchto důvodů se pro minimalizaci používá např. Nelderův-Meadův algoritmus, který k minimalizaci S nepotřebuje její derivace [10]. Více o Nelderově-Meadově algoritmu lze zjistit např. v [6]. Jestliže je inverzní úloha vedení tepla lineární, lze z Duhamelova principu odvodit přímo vztah pro q_M .

Volba parametru r je velmi důležitá. Malá hodnota r může vést k numerické nestabilitě a nežádoucím oscilacím v řešení. Je-li r příliš vysoké, tak výsledný tepelný tok bude příliš vyhlazen. Navíc se s rostoucím r zvyšuje výpočetní náročnost algoritmu. Většinou se volí r rovno 3, nebo 4. Jestliže je zvoleno $r = 1$ jedná se o tzv. Stolzovu metodu [2]. Pro odhad optimálního počtu kroků lze použít metodu popsanou v článku [11] založenou na měření rozdílu teplotní odezvy na dva po sobě jdoucí Diracovy impulsy v hustotě tepelného toku.

Jestliže je teplota v tělese měřena na více místech (jejich počet je označen J), modifikuje se účelová funkce S takto

$$S(q_M) = \sum_{i=1}^r \sum_{j=1}^J (Y_{j,M+i-1} - T_{j,M+i-1})^2,$$

kde $Y_{j,M+i-1}$ je teplota naměřená j -tým teplotním senzorem a $T_{j,M+i-1}$ je vypočtená teplota v místě j -tého teplotního senzoru.

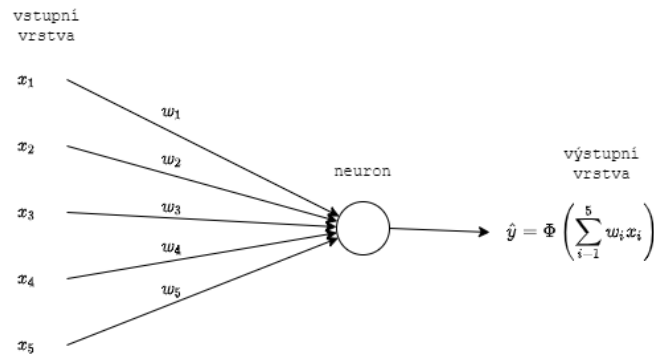
3.3 Umělé neuronové sítě

Umělé neuronové sítě jsou metodou strojového učení (což je podoblast umělé inteligence), která simuluje mechanismy učení živých organismů [1]. Lidská nervová soustava obsahuje buňky, nazývané neurony. Neurony jsou vzájemně propojeny axony a dendrity. Oblast spojení mezi axony a dendrity se nazývá synapse. Síla synaptického spojení se může měnit v důsledku vnějšího vzruchu, což je způsob, jakým se živé organismy učí.

Tento přístup je napodobován umělými neuronovými sítěmi. Výpočetní jednotky (neurony) jsou propojeny skrz váhy, které slouží stejnému účelu jako v lidské nervové soustavě. Každý vstup do neuronu je vynásoben vahou, což ovlivňuje výstup z neuronu, jak je znázorněno na obrázku 3.3. Učení umělé neuronové sítě se dosahuje změnou hodnot vah, které spojují neurony. Na rozdíl od živých organismů je však umělé neuronové síti třeba dodat pro učení páry vstupů a jim odpovídající výstupy. Váhy jsou poté upraveny tak, aby výstup umělé neuronové sítě co nejlépe odpovídal poskytnutým výstupům. Jestliže je umělé neuronové síti poskytnut dostatek učících dat, je pak schopna přesných předpovědí pro data, která k učení použita nebyla.

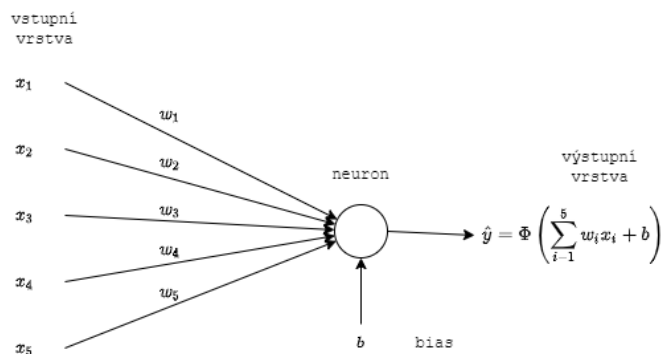
Perceptron

Perceptron je umělá neuronová síť tvořená pouze jedním neuronem, jak je znázorněno na obrázku 3.3.



Obrázek 3.3: Perceptron

Všechna učící data jsou typu (\mathbf{x}, y) , kde $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ jsou vstupy a $y \in \mathbb{R}$ výstup. Váhy neuronu jsou označeny $\mathbf{w} = (w_1, w_2, \dots, w_n)$. Výstup z neuronu se poté spočte jako $\hat{y} = \Phi(\sum_{i=1}^n w_i x_i)$. Funkce Φ se nazývá aktivační funkce. Občas je do výpočtu \hat{y} nutné zahrnout také tzv. bias b , což je konstanta, která nezávisí na vstupech. Perceptron s bias je na obrázku 3.4.



Obrázek 3.4: Perceptron s biasem

Výstup umělé neuronové sítě se pak, podle [1], vypočte jako $\hat{y} = \Phi(\sum_{i=1}^n w_i x_i + b)$.

Nechť \mathcal{D} je množina učicích dat, tj. vektorů typu (\mathbf{x}, y) . Učení perceptronu probíhá minimalizací tzv. ztrátové funkce

$$L(\mathbf{w}) = \sum_{(\mathbf{x}, y) \in \mathcal{D}} (y - \hat{y})^2.$$

K minimalizaci ztrátové funkce se používá algoritmus nazývaný stochastický gradientní sestup (stochastic gradient descent).

Aktivační funkce Φ se volí v závislosti na povaze řešené úlohy [1]. Pokud výstupem má být libovolné reálné číslo, je vhodné zvolit identitu $\Phi(x) = x$. Jestliže je výstupem pravděpodobnost nějakého jevu, je vhodné zvolit funkci, která nabývá hodnot z intervalu $\langle 0, 1 \rangle$ (např. funkci sigmoid). Pokud výstup může nabývat pouze hodnot z množiny $\{-1, 1\}$, pak je vhodné zvolit funkci $\Phi(x) = \text{sign}(x)$.

Vícevrstvé neuronové sítě

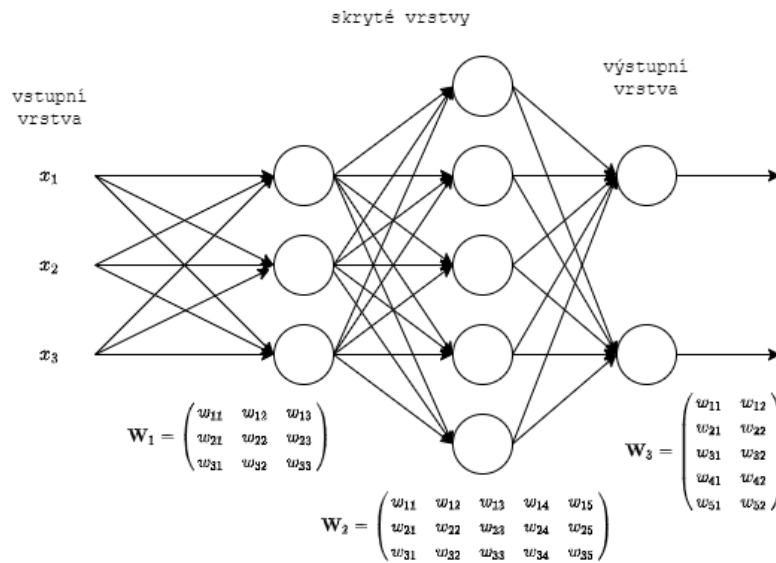
Na rozdíl od perceptronu tyto sítě obsahují více než jeden neuron. Neurony jsou uspořádány do vrstev, kterým se říká skryté, viz obrázek 3.5. Kromě skrytých vrstev má tato síť také vstupní vrstvu a vrstvu výstupní. Každý vstup je napojen do každého neuronu, což znamená, že topologie sítě je dána počtem vrstev a počtem neuronů v těchto vrstvách [1]. Opět lze uvažovat neurony s či bez bias. Nechť k označuje počet skrytých vrstev a p_i je počet neuronů v i -té vrstvě. Váhy mezi vstupní vrstvou a první skrytou vrstvou jsou uspořádány v matici \mathbf{W}_1 s rozměry $n \times p_1$. Podobně váhy v i -té skryté vrstvě jsou uspořádány do matice \mathbf{W}_i o rozměrech $p_i \times p_{i+1}$. Jestliže výstup obsahuje o neuronů, pak výstupní váhy jsou v matici \mathbf{W}_{k+1} o velikosti $p_k \times o$. Pak je výstup $\hat{\mathbf{y}}$ určen rovnicemi

$$\begin{aligned} \mathbf{h}_1 &= \Phi(\mathbf{W}_1^\top \mathbf{x}), \\ \mathbf{h}_{i+1} &= \Phi(\mathbf{W}_{i+1}^\top \mathbf{h}_i), \quad i = 1, 2, \dots, k, \\ \hat{\mathbf{y}} &= \Phi(\mathbf{W}_{k+1}^\top \mathbf{h}_k). \end{aligned}$$

K učení takových sítí se používá tzv. algoritmus zpětného šíření chyby (back-propagation algorithm). Učící množina obsahuje vektory typu $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+o}$.

Řešení inverzních úloh pomocí neuronových sítí

Pro řešení inverzních úloh vedení tepla lze využít umělé neuronové sítě. Pro využití neuronových sítí hovoří zejména jejich schopnost reprezentovat „komplexní“ zobrazení mezi



Obrázek 3.5: Neuronová síť se dvěma skrytými vrstvami

vstupní a výstupní vrstvou a také takřka neomezená možnost generování učících dat řešením přímých úloh vedení tepla [12]. Lze rozlišit dva přístupy: „Whole History Mapping“ (zkráceně WHM) a sekvenční přístup. V práci je dále popisována první ze zmíněných metod.

Bude opět uvažován příklad z kapitoly 3.1. WHM přístup je založen na tom, že vstupní vrstva umělé neuronové sítě má tolik vstupů, kolik je naměřených teplot Y_i , $i = 1, 2, \dots, N$. Výstup je hustota tepelného toku v požadovaných časech. Nevýhodou tohoto přístupu je velké množství neuronů (neboť vstupní i výstupní vrstva mají mnoho neuronů) a tedy velké množství vah, což znamená pomalé učení. Další nevýhodou pak je to, že pro jiný počet měření teploty (nebo měření v jiných časech) musí být vygenerována nová učící data a učena zcela nová síť. Tuto nevýhodu odstraňuje sekvenční algoritmus. Výhodou je, že WHM přístup je stabilní a není citlivý na šum ve vstupních datech.

Použití umělé neuronové sítě má tyto čtyři fáze:

1. Generování \mathcal{D} - data lze snadno generovat řešením přímých úloh vedení tepla. Hustotu tepelného toku je možné generovat náhodně, případně s využitím apriorní informace. Poté je vypočtena teplota v požadovaném místě a v požadovaných časech. Pro tento výpočet je možné využít libovolnou metodu pro řešení přímé úlohy vedení tepla.
2. Vytvoření sítě - topologie vstupních a výstupních vrstev je závislá na počtu naměřených teplot a počtu časových kroků, ve kterých je odhadována hustota tepelného toku. Počet neuronů skrytých vrstev je závislý na použitém učícím algoritmu.
3. Učení sítě - učící data jsou použita pro minimalizaci ztrátové funkce, učení je zastaveno ve chvíli, kdy je hodnota dostatečně malá, nebo se již nezmenšuje.
4. Testování sítě - v této fázi jsou na vstupy vložena data, která nejsou obsažena v množině \mathcal{D} .

Kapitola 4

Implementace metod

V této kapitole budou popsány autorem naprogramované funkce a skripty, které lze nalézt v příloze této práce. Většina těchto funkcí je napsána v MATLABu, pouze skript využívající neuronové sítě je napsán v Pythonu. Diskretizaci přímé stacionární úlohy vedení tepla lze provést pomocí funkce FEM. Jejím výstupem je soustava rovnic, kterou je možné řešit pomocí funkce STATIONARY. Pro nestacionární úlohy vedení tepla byly vytvořeny funkce dvě: ODE a THETA. Inverzní úlohy řeší funkce BECK, která implementuje sekvenční algoritmus, a skript WHM, který tuto úlohu řeší pomocí neuronových sítí. Obě funkce řešící inverzní úlohy vedení tepla využívají funkce, které řeší přímou úlohu vedení tepla.

4.1 Funkce řešící přímou úlohu vedení tepla

Funkce defHCEQ

Data přímé úlohy vedení tepla je možno zadat pomocí funkce defHCEQ. Vstupem této funkce je proměnná filename, což je cesta k souboru, v němž jsou uložena data o triangulaci oblasti, na které je řešena přímá úloha. Tento soubor lze získat exportem sítě z rozšíření PDE Modeller, které, mimo jiné, umožňuje vytvořit triangulaci dvourozměrných oblastí.

```
1 function eq = defHCEQ(filename)
```

Výstupem funkce je proměnná eq, která obsahuje všechna data řešené přímé úlohy. Koefficienty rovnice se zapisují jako anonymní funkce proměnných x, y, t, T, jejich označení odpovídá označení v kapitole o numerických metodách řešení přímých úloh vedení tepla.

```
6 eq.c = @(x,y,t,T) 730*(2000+110000*exp(-(T-22).^2)/1.05)); %rho*C
7 eq.p = @(x,y,t) 0.2;
8 eq.q = @(x,y,t,T) 0.0;
9 eq.g = @(x,y,t,T) 0.0;
```

Pokud funkci není předán parametr filename, pak implicitně vezme oblast definovanou v souboru regions/wall.mat. Proměnné z načteného souboru jsou uloženy do struktury eq.domain. Proměnná eq.domain.p obsahuje souřadnice uzlů triangulace, proměnná eq.domain.e obsahuje v prvních dvou řádcích indexy uzlů hran a v řádku pátém pořadové číslo hrany (to je přiděleno hraně při vytváření sítě). Proměnná eq.domain.t obsahuje trojice indexů uzlů, které tvoří prvky sítě.

```
11 if ~exist('filename','var')
```

```

12     filename = 'regions/wall';
13 end
14 load(strcat(filename, '.mat')); % Soubor s exportovanou sítí z PDE Modelleru
15 eq.domain.p = p; % Points: Uzly
16 eq.domain.e = e; % Edges: Hrany
17 eq.domain.t = t; % Triangles: Prvky

```

V další části funkce je definována proměnná `eq.time`, která obsahuje časový interval, na kterém je úloha řešena. Na dalším řádku je pak počet časových kroků, které mají být při časové diskretizaci použity.

```

19 eq.time = [0,10]; % 0, az konecny cas
20 eq.num_tsteps = 50; % Pocet casovych bodu

```

Na dalších řádcích jsou pak v proměnných `eq.f`, `eq.alpha` a `eq.beta` definované okrajové podmínky. Počet prvků v těchto polích odpovídá počtu hran oblasti, na které je řešena přímá úloha.

```

22 eq.f = {@(x,y,t) 0,@(x,y,t) 10,@(x,y,t) 0,@(x,y,t) 0};
23 eq.alpha = {@(x,y,t,T) 0,@(x,y,t,T) 0,@(x,y,t,T) 0,@(x,y,t,T) 0};
24 eq.beta = {@(x,y,t,T) 0,@(x,y,t,T) 0,@(x,y,t,T) 0,@(x,y,t,T) 0};

```

Pořadová čísla hran (pátý řádek v `eq.domain.e`), kde je předepsána Dirichletova okrajová podmínka je uložena do proměnné `eq.dir_edges`. Podobně pro Neumannovu okrajovou podmínku je použita proměnná `eq.neum_edge`.

```

25 eq.dir_edge = [2,4];
26 eq.neum_edge = [1,3];

```

Počáteční podmínka je zadána do proměnné `eq.phi`.

```

28 eq.phi = @(x,y) 10+0*x+0*y;

```

V závěru funkce se nacházejí dvě proměnné, které říkají jakého je úloha typu. Jestliže je stacionární, pak má proměnná `eq.stationary` hodnotu `true`. Podobně pro lineární problém má proměnná `eq.linear` hodnotu `true`.

```

30 eq.stationary = true;
31 eq.linear = false;
32 end

```

Funkce **STATIONARY**

Funkce je naprogramovaná pro řešení přímé stacionární úlohy vedení tepla. Vstup této funkce je proměnná `eq`, získaná voláním funkce `defHCEQ`. Nepovinný parametr `U_0` je počáteční iterace pro řešení nelineární úlohy. Pokud tento parametr při volání funkce není předán, bere se jako výchozí hodnota nulové pole. V případě lineární úlohy proměnná `U_0` není potřeba, neboť výsledná soustava rovnic je lineární. Výstupem této funkce je proměnná `U`, která obsahuje numerické řešení přímé úlohy vedení tepla v uzlech triangulace.

```

1 function [U] = STATIONARY(eq,U_0)

```

Funkce nejprve provede kontrolu, zda se jedná o stacionární úlohu. Jestliže je úloha lineární, zavolá funkci `FEM`, která sestavuje globální matici tuhosti a vektor zatížení. Poté je vyřešena příslušná soustava lineárních rovnic. Je-li úloha nelineární, provede se diskretizace opět

pomocí funkce `FEM`. V tomto případě je ovšem třeba dodat hodnotu řešení (předchozí iterace) v uzlech sítě. Výsledná soustava nelineárních rovnic je vyřešena pomocí funkce `fsolve`, která je dostupná v MATLABu. Funkce `fsolve` však předpokládá úlohu ve tvaru $\mathbf{R}(\Delta) = \mathbf{0}$, proto je soustava rovnic $\mathbf{K}(\Delta)\Delta = \mathbf{F}(\Delta)$ zapsána ve tvaru $\mathbf{R}(\Delta) = \mathbf{K}(\Delta)\Delta - \mathbf{F}(\Delta) = \mathbf{0}$.

```

7  if ~eq.stationary
8      error('Uloha neni stacionarni');
9  end
10
11  if ~exist('U_0','var')
12      U_0 = zeros(size(eq.domain.p,2),1);
13  end
14
15  if eq.linear
16      [~,K,F] = FEM(eq);
17      U = K\F;
18  else
19      U = fsolve(@(U) wrapper(U,eq),U_0);
20  end
21  end
22
23  function R = wrapper(U,eq)
24      [~,K,F] = FEM(eq,U);
25      R = K*U-F;
26  end

```

Funkce ODE

Tato funkce řeší přímou nestacionární úlohu s využitím funkce `ode23t`. Funkci ODE je nutno předat na vstupu proměnnou `eq` (viz funkce `defHCEQ`). Dále je možné předat proměnnou `t`, což je pole časových kroků. Vstupní proměnná `U0` je počáteční podmínka v každém uzlu triangulace. Jestliže `U0` při volání funkce není předána, tak se použije proměnná `eq.phi`. Jestliže je vstupní proměnná `log` nastavena na `true`, pak dělení časového intervalu je logaritmické. Výstupem funkce je pole `U`, jehož sloupce jsou řešení v uzlech triangulace. Počet sloupců je dán počtem časových kroků. Na výstupu je předáváno také použité dělení časového intervalu v proměnné `t`.

```

1  function [U,t] = ODE(eq,t,U0,log)

```

Nejprve funkce zkontroluje, že úloha není stacionární. Jestliže je pole `t` je prázdné, použije se buď rovnoměrné časové dělení, nebo logaritmická časová škála (podle proměnné `log`). Poté je inicializováno pole, do kterého bude uložen výsledek. Jestliže je pole `U0` prázdné, vyhodnotí se proměnná `eq.phi` ve všech uzlech triangulace. V prostorové oblasti je úloha diskretizována pomocí funkce `FEM`, čímž je problém převeden na soustavu obyčejných diferenciálních rovnic, která je vyřešena funkcí `ode23t`. Soustavu obyčejných diferenciálních rovnic je ovšem třeba převést na tvar $\dot{\Delta} = \mathbf{C}^{-1}(\mathbf{F} - \mathbf{K}\Delta)$.

```

8  if eq.stationary
9      error('Uloha je stacionarni. ');
10  end
11
12  if isempty(t)
13      if log
14          t = logspace(-3,log10(eq.time(2)),eq.num_tsteps-1);

```

```

15     t = [eq.time(1),t];
16     if t(end) ~= eq.time(2)
17         t(end) = eq.time(2);
18     end
19     else
20         t = linspace(eq.time(1),eq.time(2),eq.num_tsteps);
21     end
22 end
23
24 U = zeros(size(eq.phi(eq.domain.p(1,:)',eq.domain.p(2,:)),1)',size(t,2));
25 % Pocatecni podminka je zadana bud jako funkce, nebo v kazdem uzlu triangulace
26
27 if isempty(U0)
28     U(:,1) = eq.phi(eq.domain.p(1,:),eq.domain.p(2,:));
29 else
30     U(:,1) = U0;
31 end
32
33 [t,U] = ode23t(@(t,U) wrapper(t,U,eq),t,U(:,1));
34 U = U';
35 t = t';
36 end
37
38 function dUdt = wrapper(t,U,eq)
39     [C,K,F] = FEM(eq,t,U);
40     dUdt = C\ (F-K*U);
41 end

```

Funkce THETA

Funkce THETA, jako funkce ODE, řeší přímou nestacionární úlohu vedení tepla. Má stejné vstupní proměnné jako funkce ODE, navíc je ovšem předávána proměnná th , která odpovídá θ v θ -metodě. Vstupní proměnná num_impl určuje počet počátečních časových kroků, které budou řešeny implicitní Eulerovou metodou (tj. $\theta = 1$). Výstupy jsou opět proměnné U , t , stejně jako u funkce ODE.

```

1 function [U,t] = THETA(eq,t,U0,th,log,num_impl)

```

Začátek funkce je stejný jako u funkce ODE. Popis je tedy zahájen až tam, kde se funkce vzájemně liší. Nejprve je třeba zkontrolovat, že parametr th je z intervalu $\langle 0,1 \rangle$. Poté následuje cyklus od prvního do předposledního časového kroku. V každém průchodu tímto cyklem je třeba vyřešit soustavu rovnic, která vznikne diskretizací pomocí funkce FEM. Nelineární soustavy rovnic jsou opět řešeny pomocí funkce `fsolve`. Jako počáteční iterace je použita hodnota řešení v předchozím časovém kroku. Na prvních num_impl časových kroků je th nastaveno na hodnotu 1, poté se funkce vrátí k výchozí hodnotě.

```

42 if th < 0.5
43     warning('Pozor, theta metoda s th < 0.5 muze byt pro velke casove kroky
44         nestabilni!');
45 end
46 if th > 1 || th < 0
47     error('Parametr th musi byt z intervalu <0,1>!');
48 end
49 warning('off');
50 % Hodnotu parametru theta si docasne ulozim
51 th_last = th;

```

```

51 % Nastaveni fsolve parametru
52 options = optimoptions('fsolve','Display','iter','Algorithm','levenberg-
    marquardt');
53 % Theta metoda
54 if eq.linear
55     % Linearni uloha
56     for i = 1:size(t,2)-1
57         % Implicitni Eulerova metoda po num_impl kroku
58         if i < num_impl
59             th = 1;
60         else
61             th = th_last;
62         end
63         tau_i = t(i+1)-t(i);
64         t_ith = t(i)+th*tau_i;
65         [C,K,F] = FEM(eq,t_ith);
66         U(:,i+1) = (C+tau_i*th*K)\((C-tau_i*(1-th)*K)*U(:,i)+tau_i*F);
67     end
68 else
69     % Nelinearni uloha
70     for i = 1:size(t,2)-1
71         % Implicitni Eulerova metoda po num_impl kroku
72         if i < num_impl
73             th = 1;
74         else
75             th = th_last;
76         end
77         tau_i = t(i+1)-t(i);
78         t_ith = t(i)+th*tau_i;
79         [U(:,i+1),~,exitflag] = fsolve(@(U_act)wrapper(U_act,eq,tau_i,th,t_ith,U
            (:,i)),U(:,i),options);
80         if exitflag < 0
81             error('Nelinearni soustava rovnic nebyla vyresena. Zastavuji theta
                metodu.');
```

Pro lineární úlohy se více osvědčila funkce THETA, zatímco pro úlohy nelineární lze doporučit funkci ODE.

Funkce FEM

Funkce FEM provádí prostorovou diskretizaci přímé úlohy vedení tepla. Jedná se o implementaci tzv. pružinového algoritmu (viz [5]). Vstupem je proměnná eq , v proměnné t je uložen čas, ve kterém se provádí diskretizace (pro nestacionární úlohy) a v proměnné U je hodnota řešení v uzlech triangulace (pro nelineární úlohy). Výstupem této funkce jsou globální matice tepelné kapacity \mathbf{C} , matice tuhosti \mathbf{K} a vektor zatížení \mathbf{F} .

```
1 function [C,K,F] = FEM(eq,t,U)
```

Jestliže funkci nejsou při volání předány vstupní proměnné t a U, inicializuje se jejich hodnota na nulu.

```
13 if ~exist('t','var')
14     t = 0;
15 end
16 if ~exist('U','var')
17     U = zeros(size(eq.domain.p,2),1);
18 end
```

Do pomocných proměnných num_pts, num_edge a num_elem je uložen počet uzlů triangulace, počet hran a počet prvků.

```
19 num_pts = size(eq.domain.p,2);
20 num_edge = max(eq.domain.e(5,:));
21 num_elem = size(eq.domain.t,2);
```

Dále jsou do pole dir_p uloženy indexy uzlů, na kterých je předepsána Dirichletova okrajová podmínka. Podobně v poli neu_e jsou uloženy dvojice indexů uzlů, na kterých je předepsána Neumannova okrajová podmínka.

```
24 for i = 1:size(eq.dir_edge,2)
25     dir_p{i} = unique([eq.domain.e(1,eq.domain.e(5,')==eq.dir_edge(i)),eq.domain.
26                       e(2,eq.domain.e(5,')==eq.dir_edge(i))]);
27 end
28 neu_e = cell(size(eq.neum_edge,2),1); % Hraný, kde je predepsana Neumannova OP
29 for i = 1:size(eq.neum_edge,2)
30     neu_e{i} = vertcat(eq.domain.e(1,eq.domain.e(5,')==eq.neum_edge(i)),eq.domain.
31                       .e(2,eq.domain.e(5,')==eq.neum_edge(i)));
32 end
```

Nyní jsou inicializována výstupní pole C, K a F.

```
32 C = zeros(num_pts,num_pts);
33 K = zeros(num_pts,num_pts);
34 F = zeros(num_pts,1);
```

Následuje cyklus přes všechny prvky. Pro každý prvek se vypočítá lokální matice tuhosti $\mathbf{K}_e = \mathbf{K}_e^1 + \mathbf{K}_e^2$, lokální matice \mathbf{C}_e a lokální vektor zatížení \mathbf{F}_e . Lokální matice jsou poté pomocí globálního indexu prvku přiřazeny do globálních matic. Jestliže je problém stacionární, lokální matice \mathbf{C}_e se sestavuje.

```
36 for i = 1:num_elem
37     Te = eq.domain.t(1:3,i); % Prvek
38     Pe = eq.domain.p(:,Te); % Souradnice vrcholu prvku
39     Se = [mean(Pe(1,:)),mean(Pe(2,:))]; % Souradnice teziste prvku
40     de = (Pe(2,3)-Pe(2,1))*(Pe(1,2)-Pe(1,1))-(Pe(1,1)-Pe(1,3))*(Pe(2,1)-Pe(2,2));
41     re = (Pe(2,2)-Pe(2,3))*(Pe(2,3)-Pe(2,1))+(Pe(1,3)-Pe(1,2))*(Pe(1,1)-Pe(1,3));
42     se = (Pe(2,2)-Pe(2,3))*(Pe(2,1)-Pe(2,2))+(Pe(1,3)-Pe(1,2))*(Pe(1,2)-Pe(1,1));
43     te = (Pe(2,3)-Pe(2,1))*(Pe(2,1)-Pe(2,2))+(Pe(1,1)-Pe(1,3))*(Pe(1,2)-Pe(1,1));
44     Ke1 = (0.5/abs(de))*eq.p(Se(1),Se(2),t).*[-re-se, re, se; re, -re-te, te; se, te, -se
45         -te];
46     Ke2 = (abs(de)/6).*diag([eq.q(Pe(1,1),Pe(2,1),t,U(Te(1))),eq.q(Pe(1,2),Pe
47         (2,2),t,U(Te(2))),eq.q(Pe(1,3),Pe(2,3),t,U(Te(3)))]);
48     if ~eq.stationary
```

```

47     Ce = (abs(de)/6).*diag([eq.c(Pe(1,1),Pe(2,1),t,U(Te(1))),eq.c(Pe(1,2),Pe
      (2,2),t,U(Te(2))),eq.c(Pe(1,3),Pe(2,3),t,U(Te(3)))]);
48     end
49     Ke = Ke1+Ke2; % Lokalni matice K
50     Fe = (abs(de)/6).*[eq.g(Pe(1,1),Pe(2,1),t,U(Te(1)));eq.g(Pe(1,2),Pe(2,2),t,U(
      Te(2))];eq.g(Pe(1,3),Pe(2,3),t,U(Te(3)))]]; % Lokalni vektor F
51     % Prirazeni do globalni matice C, K, F
52     for j = 1:3
53         jg = eq.domain.t(j,i);
54         for k = 1:3
55             kg = eq.domain.t(k,i);
56             if ~eq.stationary
57                 C(jg,kg) = C(jg,kg) + Ce(j,k);
58             end
59             K(jg,kg) = K(jg,kg) + Ke(j,k);
60         end
61         F(jg) = F(jg) + Fe(j);
62     end
63 end

```

Cyklus přes hrany, kde je předepsána Neumannova okrajová podmínka je analogický cyklu přes trojúhelníkové prvky. „Prvkem“ je zde úsečka spojující dva uzly triangulace. Opět jsou sestaveny lokální matice tuhosti a vektor zatížení, které jsou poté jsou přiřazeny do matic globálních.

```

65     for i = eq.neum_edge
66         alpha = eq.alpha{i};
67         beta = eq.beta{i};
68         I = find(eq.neum_edge == i);
69         for j = 1:size(neu_e{I},2)
70             Pe = eq.domain.p(:,neu_e{I}(:,j)); % Souradnice prvku S
71             de = sqrt((Pe(1,2)-Pe(1,1))^2+(Pe(2,2)-Pe(2,1))^2);
72             Ks = 0.5*de.*[alpha(Pe(1,1),Pe(2,1),t,U(neu_e{I}(1,j))),0;0,alpha(Pe(1,2),
              Pe(2,2),t,U(neu_e{I}(2,j)))]]; % Lokalni matice K
73             Fs = 0.5*de.*[beta(Pe(1,1),Pe(2,1),t,U(neu_e{I}(1,j)));beta(Pe(1,2),Pe
              (2,2),t,U(neu_e{I}(2,j)))]]; % Lokalni vektor F
74             for k = 1:2
75                 kg = neu_e{I}(k,j);
76                 for l = 1:2
77                     lg = neu_e{I}(l,j);
78                     K(kg,lg) = K(kg,lg) + Ks(k,l);
79                 end
80                 F(kg) = F(kg) + Fs(k);
81             end
82         end
83     end

```

Třetí cyklus je přes uzly, na kterých je předepsána Dirichletova okrajová podmínka. Vynucení této okrajové podmínky lze, podle [5], povést tak, že do vektoru \mathbf{F} se na aktuální index (souřadnici), přiřadí velké číslo, zde označované jako kappa, vynásobené diagonálním prvkem globální matice tuhosti (na odpovídajícím indexu) a předepsanou hodnotou (tj. funkce f vyhodnocená v příslušném uzlu). Poté se tento diagonální prvek v matici \mathbf{K} vynásobí konstantou kappa. Tento postup způsobí, že hodnota řešení v uzlu bude přibližně odpovídat předepsané hodnotě.

```

85     kappa = 1e10;
86     for i = 1:size(eq.dir_edge,2)
87         for j = 1:size(dir_p{i},2)

```

```

88     jg = dir_p{i}(j);
89     Pe = eq.domain.p(:, jg);
90     F(jg) = eq.f(eq.dir_edge(i))(Pe(1), Pe(2), t) * K(jg, jg) * kappa;
91     K(jg, jg) = K(jg, jg) * kappa;
92     end
93 end
94 end

```

Skript solve - příklad použití výše popsaných funkcí

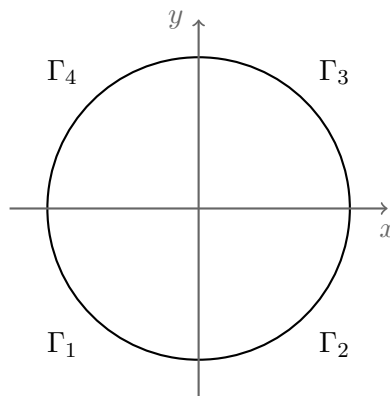
Pomocí výše popsaných funkcí bude řešena přímá stacionární úloha na jednotkovém kruhu v rovině. Rovnice vedení tepla je tvaru

$$0,2 \cdot \left[\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} \right] = 0, (x, y) \in \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}.$$

Na hranici jednotkového kruhu jsou předepsány následující okrajové podmínky

$$\begin{aligned} \frac{\partial T(x, y)}{\partial n} &= 0, (x, y) \in \Gamma_1, \\ T(x, y) &= 10, (x, y) \in \Gamma_2, \\ \frac{\partial T(x, y)}{\partial n} &= 0, (x, y) \in \Gamma_3, \\ T(x, y) &= 0, (x, y) \in \Gamma_4, \end{aligned}$$

kde $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ čtvrtiny kružnice podle obrázku 4.1 a \mathbf{n} je vnější normála.



Obrázek 4.1: Oblast úlohy řešené skriptem solve

Triangulace oblasti je uložena v příloženém souboru `regions/circle`, proto je funkci `defHCEQ` předána cesta k tomuto souboru.

```

1 clear;
2 close;
3 clc;
4 % Struktura obsahující data prime ulohy rovnice
5 eq = defHCEQ('regions/circle');

```

Pro řešení je nutno zadat funkce p, q, g , tedy jsou provedeny příkazy

```

7 eq.p = @(x,y,t) 0.2;
8 eq.q = @(x,y,t,T) 0;
9 eq.g = @(x,y,t,T) 0;

```

Okrajové podmínky jsou zadány pomocí příkazů

```

11 eq.beta{1} = @(x,y,t,T) 0;
12 eq.f{2} = @(x,y,t) 10;
13 eq.beta{3} = @(x,y,t,T) 0;
14 eq.f{4} = @(x,y,t) 0;
15 eq.neum_edge = [1,3];
16 eq.dir_edge = [2,4];

```

Jelikož se jedná o lineární stacionární úlohu jsou použity příkazy

```

18 eq.stationary = true;
19 eq.linear = true;

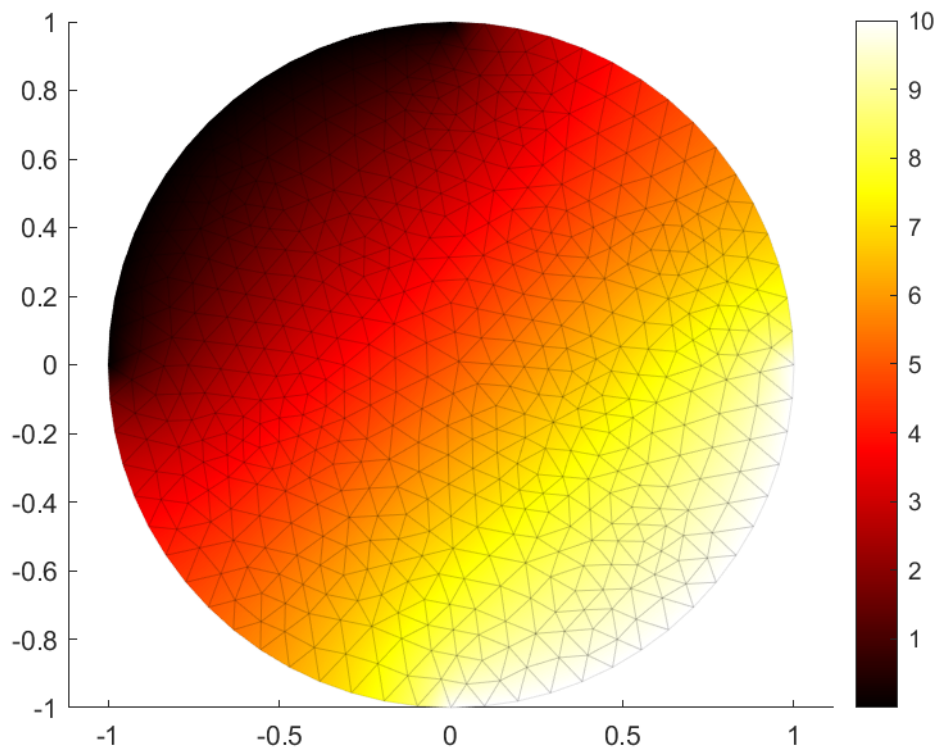
```

Nakonec je tato struktura předána funkci `STATIONARY`, která vrací řešení v proměnné `U_stationary`. To je možné si přehledně zobrazit pomocí funkce `PLOT` (viz obrázek 4.2).

```

21 [U_stationary] = STATIONARY(eq);
22 PLOT(U_stationary, [], eq, 'plots/stationary', true);

```



Obrázek 4.2: Řešení stacionární přímé úlohy vedení tepla na jednotkovém kruhu.

Dále se ve skriptu řeší nestacionární úloha s rovnicí vedení tepla

$$\frac{\partial T(x, y, t)}{\partial t} = 0,2 \cdot \left[\frac{\partial^2 T(x, y, t)}{\partial x^2} + \frac{\partial^2 T(x, y, t)}{\partial y^2} \right], (x, y) \in \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}, t \in (0, 10).$$

Okrajové podmínky jsou stejné jako ve stacionárním případě. Zbývá tedy předepsat počáteční podmínku

$$T(x, y, 0) = 10.$$

Ve struktuře eq je tedy potřeba nastavit několik dalších proměnných.

```
24 eq.stationary = false;
25 eq.linear = true;
26 eq.c = @(x,y,t,T) 1;
27 eq.time = [0,10];
28 eq.num_tsteps = 50;
29 eq.phi = @(x,y) 10+0*x+0*y;
```

Nyní již lze předat proměnnou eq funkci ODE, případně THETA.

```
31 [U_ode,t_ode] = ODE(eq, [], [], true);
32 PLOT(U_ode,t_ode,eq, 'plots/nonstationary_ode', true);
```

```
34 [U_theta,t_theta] = THETA(eq, [], [], 0.5, true, 5);
35 PLOT(U_theta,t_theta,eq, 'plots/nonstationary_theta', true);
```

Funkce PLOT v tomto případě zobrazí a uloží animaci, která se skládá z hodnot řešení v časových krocích. Pro tento konkrétní příklad si lze animace prohlédnout v příložených souborech plots/nonstationary_ode a plots/nonstationary_theta.

4.2 Funkce řešící inverzní úlohy vedení tepla

Funkce defIHCP

Data inverzní úlohy je možno zadat pomocí funkce defIHCP. Vstupem do této funkce je cesta k souboru, ve kterém jsou data o měření teploty v tělese (čas měření a teplota) a data o podkladové přímé úloze.

```
1 function ihcp = defIHCP(filename,eq)
```

Ručně jsou poté zadávány počty teplotních senzorů, pořadové číslo hrany, kde je hledaná hustota tepelného toku a pozice teplotních senzorů.

```
6 ihcp.num_sensor = 1; % Pocet teplotnich senzorů
7 ihcp.eq = eq; % Data přímé úlohy vedení tepla
8 ihcp.ind_edge = 4; % Index hrany, na které budeme odhadovat tepelný tok
9 ihcp.location = [0.01,0.5]; % Pozice teplotnich senzorů - [x y]
10 load(strcat(filename, '.mat'));
11 ihcp.t = t; % Casove zmkanky
12 ihcp.T = T; % Namerene teploty
13 end
```

Výstupem této funkce je proměnná obsahující data inverzní úlohy vedení tepla. Vývoj teploty v jednom, případně více bodech, lze z řešení přímé úlohy získat a uložit pomocí funkce INTERPOLATE.

Funkce BECK

Tato funkce je implementace sekvenčního algoritmu. Vstupní proměnná `ihcp` obsahuje data řešené inverzní úlohy (viz funkce `defIHCP`). Vstupní proměnná `r` představuje počet použitých dopředných kroků.

```
1 function [q,t_temp] = BECK(ihcp,r)
```

Nejprve je inicializován prázdný vektor q , do kterého bude přidávána odhadnutá hustota tepelného toku v aktuálním časovém kroku. Následuje inicializace pomocné proměnné $U0$, do které jsou uloženy hodnoty počáteční podmínky v uzlech triangulace. Poté následuje hlavní cyklus funkce. Při prvních průchodu je jako počáteční podmínka použita proměnná $U0$, poté je počáteční teplota vypočítána z již odhadnutých hodnot hustoty tepelného toku q . Minimalizace je prováděna pomocí funkce `fminsearch` (jedná se o Nelderovu-Meadovu metodu). Jako výchozí hodnotu pro minimalizaci je použita hustota tepelného toku v předchozím časovém kroku. Odhadnutá hustota tepelného toku je poté přidána k vektoru q a pokračuje se dalším časovým krokem.

```
4 q = [];  
5 options = optimset('Display','iter');  
6 U0 = ihcp.eq.phi(ihcp.eq.domain.p(1,:),ihcp.eq.domain.p(2,:));  
7 for M = 1:size(ihcp.t,2)-r  
8     fprintf('BSA: Krok %i z %i.\n',M,size(ihcp.t,2)-r);  
9     if M > 1  
10        t_temp = ihcp.t(1:M);  
11        ihcp.eq.beta{ihcp.ind_edge} = @(x,y,t,T) vect2fun(q,t_temp,t);  
12        if ihcp.eq.linear  
13            [U,~] = THETA(ihcp.eq,t_temp,[],0.5,false,5);  
14        else  
15            [U,~] = ODE(ihcp.eq,t_temp,[],false);  
16        end  
17    else  
18        U = U0;  
19    end  
20    q0 = 0;  
21    if M > 1  
22        q0 = q(M-1);  
23    end  
24    q = [q,fminsearch(@(qM) wrapper(qM,M,ihcp,r,U(:,end)),q0,options)];  
25 end  
26 end
```

Funkce `wrapper` obaluje účelovou funkci ve funkci `BECK`. Jejím účelem je výpočet teploty v daných bodech po r budoucích časových kroků, jestliže hustota tepelného toku bude mít konstantní hodnotu qM . Pomocí funkce `INTERPOLATE` je obdržena teplota v bodech teplotních senzorů. Poté je spočtena hodnota součtu kvadrátů odchylek od teplot naměřených, což je výstupem této funkce.

```
28 function S = wrapper(qM,M,ihcp,r,U0)  
29     t_temp = ihcp.t(M:M+r-1);  
30     q_temp = qM*ones(1,r);  
31     ihcp.eq.beta{ihcp.ind_edge} = @(x,y,t,T) vect2fun(q_temp,t_temp,t);  
32     if ihcp.eq.linear  
33         [T,~] = THETA(ihcp.eq,t_temp,U0,0.5,false,0);  
34     else  
35         [T,~] = ODE(ihcp.eq,t_temp,U0,false);  
36     end
```

```

37 [T,~] = INTERPOLATE(ihcp.eq,T,t_temp,ihcp.location,false,'');
38 S = sum(sum((ihcp.T(:,M:M+r-1)-T).^2));
39 end

```

Skript WHM

Tento skript využívá k odhadu hustoty tepelného toku neuronové sítě. Je psaný v programovacím jazyce Python. Nejprve je potřeba nainportovat použité knihovny (jestliže uživatel nemá knihovny nainstalované, lze použít prostředí Google Colab, ve kterém se instalovat nemusejí).

```

import pandas as pd
import numpy as np
import tensorflow as tf
import io
import matplotlib.pyplot as plt
from google.colab import files
from sklearn.preprocessing import StandardScaler

```

Model neuronové sítě je vytvořen pomocí knihovny TensorFlow. Tato neuronová síť má vstupní vektor velikosti 50, má 20 neuronů ve skryté vrstvě a výstupní vektor má také velikost 50, jak bylo popsáno v kapitole o neuronových sítích.

```

inputs = tf.keras.Input(shape=50, name='inputs')
x = tf.keras.layers.Dense(20, activation='sigmoid', name='hidden')(inputs)
outputs = tf.keras.layers.Dense(50, activation='linear', name='outputs')(x)
model = tf.keras.Model(inputs=inputs, outputs=outputs, name='
    WholeHistoryMappingModel')
model.summary()
tf.keras.utils.plot_model(model, 'model.png', show_shapes=True)

```

Poté jsou načtena učící data. Autorem použitá data je možno nalézt v příloze práce. Pro generování dalších učících dat lze použít autorem naprogramovaný a k práci přiložený skript `generate_training_data`.

```

training_set = files.upload()

```

Následující řádky kódu zajistí, aby byla data uspořádána do podoby vhodné pro učení neuronové sítě. Jedná se o pole z knihovny numpy, které musí mít rozměr $50 \times n$, kde n je počet načtených souborů. Dále je využita normalizace výstupního vektoru.

```

from matplotlib import axes
frames = list()
n = 0
for file in training_set:
    frame = pd.read_csv(io.BytesIO(training_set[file]), header=None, delimiter=';')
    if frame.shape == (50,2):
        n = n + 1
        frames.append(frame)
data = pd.concat(frames)
data.rename(columns={0:'temperature', 1:'heat flux'}, inplace=True)
labels = data.pop('heat flux').to_numpy()
features = data.pop('temperature').to_numpy()
x_train = np.reshape(features, (n,50)).astype('float32')
y_train = np.reshape(labels, (n,50)).astype('float32')

```

```

scaler = StandardScaler()
scaler.fit(y_train.T)
y_train = scaler.transform(y_train.T).T

```

Následuje kompilace modelu a jeho učení, opět pomocí knihovny TensorFlow. Parametry funkce `fit` je možné podle potřeby upravovat, aby bylo dosaženo co nejmenší hodnoty ztrátové funkce.

```

model.compile(loss=tf.keras.losses.mean_squared_error, optimizer=tf.keras.
    optimizers.Adam(0.001))
history = model.fit(x_train, y_train, epochs=150, batch_size=5, validation_split
    =0.2)
plt.title('Loss / Mean Squared Error')
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')

```

Po ukončení učení jsou načteny testovací soubory.

```

testing_set = files.upload()

```

Poté je model použit k předpovědi a výsledky jsou srovnány pomocí grafu.

```

frame = pd.read_csv('triangularT_a=2_b=5_c=8_M=500000.csv', header=None,
    delimiter=';')
frame.rename(columns={0:'temperature', 1:'heat flux'}, inplace=True)
test_labels = frame.pop('heat flux')
test_features = frame.pop('temperature')
x_test = test_features.to_numpy()
x_test = np.reshape(x_test, (1,50))
y_real = test_labels.to_numpy()
y_real = np.reshape(y_real, (1,50))
y_predicted = model.predict(x_test)
time_steps = np.linspace(0, 10, 50)
print(np.mean(scaler.mean_))
print(np.mean(scaler.var_))
y_predicted = y_predicted * np.sqrt(np.mean(scaler.var_)) + np.mean(scaler.mean_
    )
plt.scatter(time_steps, y_predicted, color='red')
plt.scatter(time_steps, y_real, color='blue')

```

Model a dosažené výsledky je možné uložit.

```

from google.colab import drive
drive.mount('/content/gdrive')
model.save('/content/gdrive/MyDrive/Models/WHM_model_tri_200_01')
pd.DataFrame(y_predicted).to_csv('/content/gdrive/MyDrive/Models/q_tri_2.csv')

```

Kapitola 5

Dosažené výsledky a porovnání metod

V této kapitole bude popsána autorem řešená inverzní úloha vedení tepla se změnou fáze a dosažené výsledky, které budou kvantifikovány pomocí střední kvadratické chyby. V závěru kapitoly se pak nachází zhodnocení použitých metod a jejich porovnání.

5.1 Řešená inverzní úloha

Materiál se změnou fáze

Pro vytvoření numerického modelu materiálu se změnou fáze byla použita metoda efektivní tepelné kapacity. Dále bude uvažován materiál s fyzikálními vlastnostmi z tabulky 5.1.

označení	název	hodnota	jednotka
ρ	hustota	730	$[\text{kg} \cdot \text{m}^{-3}]$
λ	součinitel tepelné vodivosti	0,2	$[\text{W} \cdot \text{m}^{-1} \cdot ^\circ\text{C}^{-1}]$
c	tepelná kapacita	$2 + 110 \cdot e^{-\frac{(T-22)^2}{1,05}}$	$[\text{kJ} \cdot \text{kg}^{-1}]$

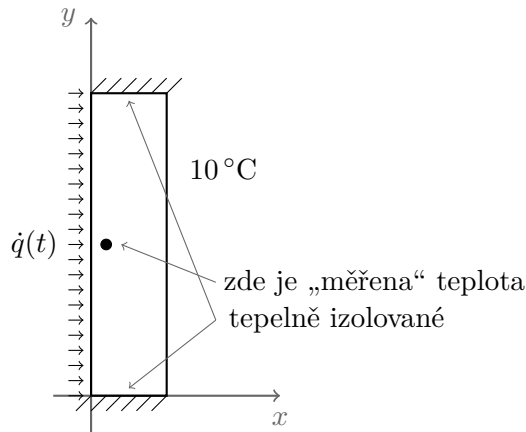
Tabulka 5.1: Fyzikální vlastnosti materiálu se změnou fáze

Tvar tělesa a okrajové podmínky

Prostorová oblast je tvaru obdélníku s rozměry $1 \text{ m} \times 0,05 \text{ m}$. Horní a spodní konec jsou tepelně izolovány. Na pravém okraji je předepsána teplota 10°C . Na levém okraji je předepsána hustota tepelného toku jako funkce času $\dot{q} = \dot{q}(t)$, který bude inverzní úlohou odhadován. Celkem jsou okrajové podmínky tvaru

$$\begin{aligned} -\lambda \frac{\partial T}{\partial n}(0, y, t) &= \dot{q}(t), \\ T(0, 05; y; t) &= 10, \\ \frac{\partial T}{\partial n}(x, 0, t) &= \frac{\partial T}{\partial n}(x, 1, t) = 0, \end{aligned}$$

kde $(x, y) \in (0; 0,05) \times (0, 1)$, $t \in (0, 10)$. Tvar tělesa a okrajové podmínky jsou zobrazeny na obrázku 5.1. Počáteční teplota tělesa je $T(x, y, 0) = 10^\circ\text{C}$. Úloha je uvažována na intervalu 0 s až 10 s.



Obrázek 5.1: Inverzní úloha

Neznámá okrajová podmínka a průběh teploty

Byly použity dva průběhy hustoty tepelného toku, první z nich je po částech lineární funkce, druhý z nich je po částech konstantní funkce

$$\dot{q}_1(t) = \begin{cases} \frac{M}{3}t - \frac{2M}{3} & t \in \langle 2, 5 \rangle, \\ -\frac{M}{3} + \frac{8M}{3} & t \in (5, 8), \\ 0 & \text{jinak,} \end{cases}$$

$$\dot{q}_2(t) = \begin{cases} M & t \in \langle 2, 5 \rangle, \\ 0 & \text{jinak,} \end{cases}$$

kde $M = 5 \cdot 10^5$. Pro \dot{q}_1 , \dot{q}_2 byly, pomocí funkcí popsaných v předchozí kapitole, vyřešeny přímé úlohy vedení tepla. Prostorová diskretizace byla provedena pomocí metody konečných prvků (tedy funkce FEM), bylo použito 297 uzlů. Časový krok byl 0,2 s a k časové diskretizaci byla použita funkce ODE. Jako vstup pro řešení inverzních úloh byla použita vypočtená teplota v bodě (0,01; 0,5) m.

Řešení inverzní úlohy pomocí funkce BECK

Ve skriptu `solve_ihcp` se nachází řešení výše popsané inverzní úlohy pomocí funkce BECK. Nejprve jsou vložena data přímé úlohy, která je vyřešena pomocí funkce ODE.

```

4 % Struktura obsahující data přímé úlohy
5 eq = defHCEQ();
6 eq.c = @(x,y,t,T) 730*(2000+110000*exp(-((T-22).^2)/1.05));
7 eq.p = @(x,y,t) 0.2;
8 eq.q = @(x,y,t,T) 0;
9 eq.g = @(x,y,t,T) 0;
10 % Okrajové podmínky
11 eq.beta{1} = @(x,y,t,T) 0;
12 eq.f{2} = @(x,y,t) 10;
13 eq.beta{3} = @(x,y,t,T) 0;
14 M=5e5;
15 eq.beta{4} = @(x,y,t,T) (M*t/3-2*M/3).*(t > 2 & t < 5)+(-M*t/3+8*M/3).*(t >= 5 &
    t < 8);
16 %eq.beta{4} = @(x,y,t,T) (M).*(t >= 2 & t <= 5);

```

```

17 eq.neum_edge = [1,3,4];
18 eq.dir_edge = [2];
19 % Dalsi parametry
20 eq.stationary = false;
21 eq.linear = false;
22 eq.time = [0, 10];
23 eq.num_tsteps = 50;
24 eq.phi = @(x,y) 10+0*x+0*y;
25 [U,t] = ODE(eq, [], [], false);
26 PLOT(U,t,eq,'plots/nonstationary_ode', false);

```

Průběh teploty v čase v bodě je uložen do souboru nonlinear.

```

27 % Ulozeni prubehu teploty v jednom bode
28 [T,t] = INTERPOLATE(eq,U,t,[0.01,0.5],true,'ihcp_data/nonlinear');

```

Dále je volána funkce defIHCP a do struktury, kterou vrátí jsou uložena data inverzní úlohy vedení tepla.

```

30 % Data inverzni ulohy
31 ihcp = defIHCP('ihcp_data/nonlinear',eq);
32 ihcp.num_sensor = 1;
33 ihcp.eq = eq;
34 ihcp.ind_edge = 4;
35 ihcp.location = [0.01,0.5];

```

Nyní je již konečně volána funkce BECK se třemi dopřednými kroky.

```

36 [q,t] = BECK(ihcp,3);
37 MSE = EVALUATE(q,t,[],[],eq.beta{4},[0 10],'plots/inverse_linear','Sekvencni
    algoritmus',[]);

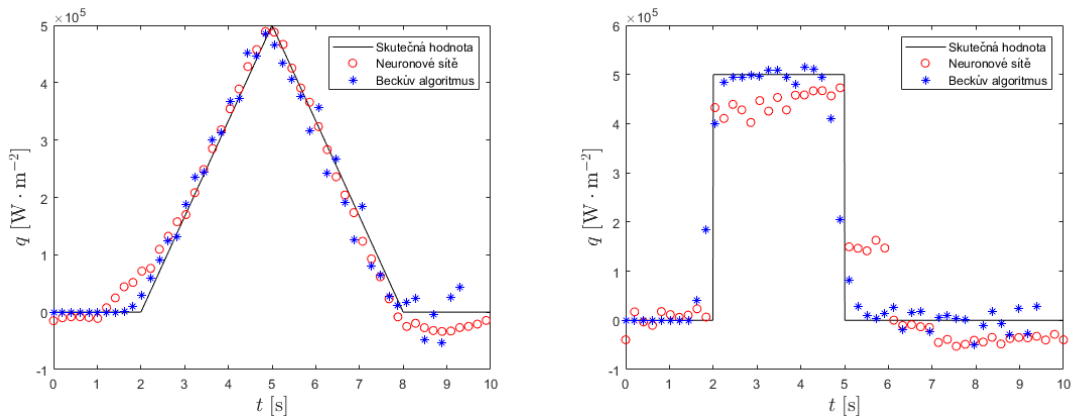
```

Generování učicích dat pro neuronové sítě

Pro řešení pomocí neuronových sítí bylo třeba vygenerovat dostatečný počet odpovídajících si průběhů hustoty tepelného toku a teploty. Pomocí přiloženého skriptu `generate_training_data` bylo vygenerováno zhruba 250 těchto dvojic pro oba případy. Nejprve byl náhodně vygenerován průběh tepelného toku stejného typu, tedy po částech lineární a po částech konstantní. Poté byla, opět pomocí funkcí z předchozí kapitoly, vyřešena přímá úloha vedení tepla a potřebná data uložena do souborů, které jsou přiloženy k práci.

5.2 Výsledky a porovnání metod

Odhadnuté a skutečné hustoty tepelných toků pomocí sekvencní metody a neuronových sítí lze vidět na obrázku 5.2. Výsledky řešení této úlohy byly autorem publikovány v příspěvku [9] již během zpracování diplomové práce a budou prezentovány na mezinárodní konferenci PRES'22, která se bude konat v září 2022 v Bolu v Chorvatsku.



Obrázek 5.2: Výsledky Beckova sekvenčního algoritmu a umělé neuronové sítě

Výsledky obou metod byly kvantifikovány pomocí SMSE (scaled mean squared error). Ta se vypočte pomocí vztahu $SMSE = \frac{10^9}{n} \sum_1^n (q_i - q_i^*)$, kde q_i je hodnota hustoty tepelného toku odhadnutá řešením inverzní úlohy a q_i^* je skutečná hodnota hustoty tepelného toku v čase t_i . Čím nižší je tato hodnota, tím lépe jsou hodnoty hustoty tepelného toku odhadnuty. Hodnoty SMSE pro řešené příklady na obrázku 5.2 jsou uvedeny v tabulce 5.2.

	BECK	WHM
\dot{q}_1	0,66	0,58
\dot{q}_2	3,36	3,91

Tabulka 5.2: SMSE pro sekvenční algoritmus a neuronové sítě a případy \dot{q}_1 , \dot{q}_2 ve $[W^2 \cdot m^{-4}]$

Obě metody dosahují pro oba případy srovnatelných výsledků. Pro hustotu tepelného toku \dot{q}_1 byly neuronové sítě o trochu lepší než sekvenční metoda, pro \dot{q}_2 tomu bylo naopak. V případě hustoty tepelného toku \dot{q}_2 jsou hodnoty SMSE vyšší než pro \dot{q}_1 , což je způsobené nespojitostí \dot{q}_2 .

Pro hodnocení těchto metod je ovšem potřeba zvážit jejich další klady a zápory. V případě Beckova sekvenčního algoritmu nebyla poskytnuta žádná apriorní informace o hustotě tepelného toku, jeho odhad touto metodou je však velmi časově náročný (zhruba 4-5 hodin pro případy výše). Navíc je třeba určit počet dopředných kroků r , což se většinou provádí metodou pokus-omyl. Při generování učicích dat byla použita apriorní informace o hustotě tepelného toku. Samotné učení neuronové sítě je pak otázkou zhruba 30 minut. Nicméně při malé změně dat úlohy by bylo potřeba generovat nová učicí data, což v případě sekvenční metody není potřeba. Také je třeba jistě zkušenosti s učení neuronové sítě, neboť snadno může dojít k jevu, který se nazývá „overfitting“. Tento jev je charakterizován tím, že neuronová síť funguje skvěle na učicí data, avšak ztratí schopnost správné předpovědi pro data, která v učicí množině nejsou.

Závěr

Úlohy výpočtového řešení přenosu tepla často vyžadují zpětné stanovení počátečních nebo okrajových podmínek, případně fyzikálních vlastností materiálů na základě vývoje daného systému v čase. Cílem této diplomové práce bylo seznámit se s problematikou inverzních úloh přenosu tepla, vytvořit numerický model přenosu tepla systému s fázovou změnou. Dalším cílem bylo implementovat zvolené matematické metody pro řešení inverzní úlohy přenosu tepla a tyto metody aplikovat na testovací úlohy a vzájemně porovnat jejich chování.

V první kapitole práce je pozornost zaměřena na mechanismy přenosu tepla a jejich matematické modelování. Dále je formulována přímá úloha vedení tepla - tedy úloha najít funkci, která splňuje rovnici vedení tepla na dané oblasti, okrajové podmínky a počáteční podmínku. V případě ustáleného vedení tepla se jedná o okrajový problém pro eliptickou parciální diferenciální rovnici, v nestacionárním případě o počátečně-okrajový problém pro parabolickou parciální diferenciální rovnici, v obou případech je neznámou teplota. Modelování fázové změny je uvažováno pomocí metody efektivní tepelné kapacity a metody entalpie, první z uvedených metod vede na nelineární přímou úlohu přenosu tepla.

Kapitola druhá se věnuje výpočtovému řešení přímých úloh přenosu tepla. Stacionární úloha je diskretizována metodou konečných prvků. Nestacionární úlohy jsou řešeny metodou přímek. Nejprve je opět úloha diskretizována metodou konečných prvků, čímž je úloha převedena na počáteční problém pro soustavu obyčejných diferenciálních rovnic. Tato soustava je poté řešena θ -metodou. Diskretizace nelineárních úloh probíhá formálně stejně jako v lineárním případě, výsledné soustavy algebraických rovnic jsou však nelineární a řeší se vhodnými numerickými metodami.

Třetí kapitola se věnuje formulaci inverzní úlohy přenosu tepla pro neznámou okrajovou podmínku. Dále jsou uvedeny dvě metody řešení této úlohy: osvědčená Beckova sekvenční metoda a řešení pomocí umělých neuronových sítí. Obě metody je možné použít pro řešení lineárních i nelineárních úloh.

V následující kapitole jsou popsány autorem práce implementované metody. Sekvenční metoda i umělé neuronové sítě vyžadují efektivní řešení velkého počtu přímých úloh. Nejprve jsou tedy popsány funkce, které tyto úlohy řeší a na jednoduchém příkladě je ukázáno jejich použití. Dále je popsána implementace Beckova sekvenčního algoritmu, který předešlé funkce využívá. Dále je uveden skript, který umožňuje řešit inverzní úlohu pomocí umělých neuronových sítí. Pro použití neuronových sítí je ovšem třeba vytvořit tzv. učící data, čehož je docíleno řešením velkým počtu přímých úloh. Poté následuje fáze učení, kdy je minimalizována tzv. ztrátová funkce. Poté je již možno použít neuronovou síť pro predikci.

V poslední kapitole autor formuluje inverzní úlohu přenosu tepla s fázovou změnou. Fázová změna je modelována pomocí metody efektivní tepelné kapacity, inverzní úloha je tedy nelineární. Na tuto úlohu jsou poté aplikovány obě zvolené metody. Součástí práce jsou zdrojové kódy pro řešení sekvenčním algoritmem a také pro generování učících dat pro neuronovou síť. Účinnost metod byla kvantifikována pomocí SMSE (scaled mean squared

error). Byly zvoleny dva testovací průběhy hustoty tepelného toku: spojitý, po částech lineární a nespojitý, po částech konstantní průběh. Obě metody dosahují pro oba průběhy srovnatelných výsledků. V případě nespojitého průběhu je hodnota SMSE vyšší než v případě spojitého průběhu. V závěru kapitoly jsou pak metody srovnány z „uživatelského hlediska“.

Seznam použitých zdrojů

- [1] AGGARWAL, Charu C. *Neural networks and deep learning: a textbook*. Cham: Springer, [2018]. ISBN 978-3-319-94462-3.
- [2] BECK, James V., Ben BLACKWELL a Charles R. St. CLAIR, JR. *INVERSE HEAT CONDUCTION: Ill-posed Problems*. New York: Wiley, c1985. ISBN 0-471-08319-4.
- [3] CRANK, John. *Free and moving boundary problems*. Oxford: Clarendon Press, [1984]. Oxford science publications. ISBN 0-19-853370-5.
- [4] ČERMÁK, Libor. *Numerické metody pro řešení diferenciálních rovnic*. Brno: Litera Brno, 2013. ISBN 978-80-903586-7-6.
- [5] ČERMÁK, Libor. *Algoritmy metody konečných prvků*. Brno: PC-DIR Real, 2000. Učební texty vysokých škol. ISBN 80-214-1697-1.
- [6] ČERMÁK, Libor a Rudolf HLAVIČKA. *Numerické metody*. Vydání třetí. Brno: Akademické nakladatelství CERM, 2016. ISBN 978-80-214-5437-8.
- [7] HAHN, David W. a M. Necati ÖZİŞİK. *Heat conduction*. Third edition. New Jersey: Wiley, 2012. ISBN 978-0-470-90293-6.
- [8] CHARVÁT, Pavel, Lubomír KLIMEŠ a Martin ZÁLEŠÁK. *Utilization of an Air-PCM Heat Exchanger in Passive Cooling of Buildings: A Simulation Study on the Energy Saving Potential in Different European Climates*. *Energies* [online]. 2019, 12(6) [cit. 2022-05-16]. ISSN 1996-1073. Dostupné z: doi:10.3390/en12061133
- [9] KLIMEŠ, Lubomír, Petr KAMARÝT, Pavel CHARVÁT, Martin ZÁLEŠÁK a Martin PEŠEK. *Solution to inverse heat transfer problems by means of soft computing approach and its comparison to the well-established Beck's method*. *Chemical Engineering Transactions*, Vol. 94, 2022, in press. ISSN 2283-9216.
- [10] KOMÍNEK, Jan. *Pokročilé metody pro inverzní úlohy vedení tepla*. Brno, 2018. Dizertační práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Laboratoř přenosu tepla a proudění. Vedoucí práce Prof. Ing. Miroslav Raudenský, CSc.
- [11] KOMÍNEK, Jan a Michal POHANKA. *Estimation of the number of forward time steps for the sequential Beck approach used for solving inverse heat-conduction problems*. *Materiali in tehnologije*. 2016, 50(2), 207-210. ISSN 15802949. Dostupné z: doi:10.17222/mit.2014.192
- [12] KREJSA, J., K. A. WOODBURY, J. D. RATLIFF a M. RAUDENSKY. *Assessment of strategies and potential for neural networks in the inverse heat conduction problem*.

Inverse Problems in Engineering. 1999, 7(3), 197-213. ISSN 1068-2767. Dostupné z: doi:10.1080/174159799088027694

- [13] LIU, Shuli, Yongcai LI a Yaqin ZHANG. *Mathematical solutions and numerical models employed for the investigations of PCMs' phase transformations*. Renewable and Sustainable Energy Reviews [online]. 2014, 33, 659-674 [cit. 2022-05-16]. ISSN 13640321. Dostupné z: doi:10.1016/j.rser.2014.02.032
- [14] PAVELEK, Milan. *Termomechanika*. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4300-6.

Seznam symbolů

Symbol	Jednotka	Název veličiny
\mathbf{x}	[m]	bod v prostoru $\mathbb{R}^3, \mathbb{R}^2$
t	[s]	čas
$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^\top$	[–]	gradient
$\partial\Omega$	[–]	hranice oblasti Ω
ρ	[kg · m ⁻³]	hustota
ρ_l	[kg · m ⁻³]	hustota kapalné fáze
ρ_s	[kg · m ⁻³]	hustota pevné fáze
$\dot{\mathbf{q}}$	[W · m ⁻²]	hustota tepelného toku
g	[W · m ⁻³]	hustota výkonu zdrojů tepla
V	[–]	kontrolní objem
T_1, T_2	[K], [°C]	krajní body intervalu teplot, ve kterém dochází ke změně fáze
$\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}$	[K · m ⁻²]	Laplaceův operátor
L	[J · kg ⁻¹]	latentní teplo fázové změny
H	[J · kg ⁻¹]	měrná entalpie
c	[J · kg ⁻¹ · K ⁻¹]	měrná tepelná kapacita
c_l	[J · kg ⁻¹ · K ⁻¹]	měrná tepelná kapacita kapalné fáze
c_s	[J · kg ⁻¹ · K ⁻¹]	měrná tepelná kapacita pevné fáze
\mathbf{n}	[–]	normála
Ω	[–]	oblast v $\mathbb{R}^3, \mathbb{R}^2$
s	[m]	poloha rozhraní mezi fázemi
ε	[–]	poměrná zářivost
h	[W · m ⁻² · K ⁻¹]	součinitel přestupu tepla
λ	[W · m ⁻¹ · K ⁻¹]	součinitel tepelné vodivosti
λ_l	[W · m ⁻¹ · K ⁻¹]	součinitel tepelné vodivosti kapalné fáze
λ_s	[W · m ⁻¹ · K ⁻¹]	součinitel tepelné vodivosti pevné fáze
a	[m ² · s ⁻¹]	součinitel teplotní vodivosti
x, y, z	[m]	souřadnice v kartézské soustavě souřadnic
$\sigma = 5,669 \cdot 10^{-8}$	[W · m ⁻² · K ⁴]	Stefanova-Boltzmannova konstanta
T	[K], [°C]	teplota
T_l	[K], [°C]	teplota kapalné fáze
T_s	[K], [°C]	teplota pevné fáze

Symbol	Jednotka	Název veličiny
$Y_{j,i}$	[K], [°C]	teplota v místě j v čase t_i
T_m	[K], [°C]	teplota změny fáze
$\nabla T = (\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z})^\top$	[K · m ⁻¹]	teplotní gradient
\dot{q}	[W · m ⁻²]	velikost hustoty tepelného toku

Seznam obrázků

1.1	Hodnoty součinitele tepelné vodivosti různých látek podle [7]	10
1.2	Závislost součinitele tepelné vodivosti na teplotě [7] (u obou obrázků byly přeloženy popisky)	11
1.3	Příklad přímé úlohy vedení tepla	15
1.4	Tuhnutí kapaliny	16
1.5	Závislost entalpie na teplotě	18
2.1	Příklad triangulace čtverce	21
2.2	Příklad očíslování trojúhelníků a uzlů	23
2.3	Lineární průběh Δ na intervalu $\langle t_n, t_{n+1} \rangle$	26
3.1	Inverzní úloha	28
3.2	M -tý krok sekvenčního algoritmu	31
3.3	Perceptron	32
3.4	Perceptron s biasem	33
3.5	Neuronová síť se dvěma skrytými vrstvami	34
4.1	Oblast úlohy řešené skriptem <code>solve</code>	42
4.2	Řešení stacionární přímé úlohy vedení tepla na jednotkovém kruhu.	43
5.1	Inverzní úloha	49
5.2	Výsledky Beckova sekvenčního algoritmu a umělé neuronové sítě	51