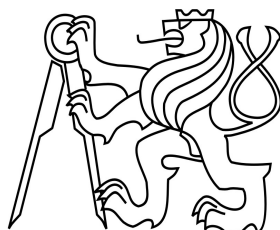


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební

Katedra betonových a zděných konstrukcí



Studijní program: Stavební inženýrství

Studijní obor: Integrovaná bezpečnost staveb

133 DPM – Diplomová práce

**POSOUZENÍ POŽÁRNÍ ODOLNOSTI
ŽELEZOBETONOVÝCH DESEK POMOCÍ
RŮZNÝCH VÝPOČETNÍCH METOD**

FIRE RESISTANCE ASSESSMENT OF REINFORCED CONCRETE SLABS
USING DIFFERENT CALCULATION METHODS

Bc. Veronika Pražáková

vedoucí práce: Ing. Radek Štefan, Ph.D.

2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Pražáková Jméno: Veronika Osobní číslo: 458566
Zadávající katedra: K133 - Katedra betonových a zděných konstrukcí
Studijní program: Stavební inženýrství
Studijní obor: Integrální bezpečnost staveb

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Posouzení požární odolnosti železobetonových desek pomocí různých výpočetních metod

Název diplomové práce anglicky: Fire resistance assessment of reinforced concrete slabs using different calculation methods

Pokyny pro vypracování:

Rešerše literatury

Popis vybraných metod

Tvorba výpočetního programu

Řešené příklady

Vyhodnocení

Závěr

Seznam doporučené literatury:

ČSN EN 1991-1-2 Eurokód 1: Zatížení konstrukcí - Část 1-2: Obecná zatížení - Zatížení konstrukcí vystavených účinkům požáru

ČSN EN 1992-1-1 Eurokód 2: Navrhování betonových konstrukcí - Část 1-1: Obecná pravidla a pravidla pro pozemní stavby

ČSN EN 1992-1-2 Eurokód 2: Navrhování betonových konstrukcí - Část 1-2: Obecná pravidla - Navrhování konstrukcí na účinky požáru

Procházka, J., Štefan, R., Vašková, J. Navrhování betonových a zděných konstrukcí na účinky požáru.

Jméno vedoucího diplomové práce: Ing. Radek Štefan, Ph.D.

Datum zadání diplomové práce: 28.9.2020

Termín odevzdání diplomové práce: 3.1.2021

Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku

Podpis vedoucího práce

Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

Datum převzetí zadání

Podpis studenta(ky)

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a že jsem uvedla všechny použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Nemám námitek proti použití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v aktuálním znění (15.02.2019).

V Praze dne 03.01.2020

.....

Podpis

Poděkování

Chtěla bych poděkovat Ing. Radku Štefanovi, Ph.D., za cenné rady a připomínky a vstřícnost při konzultacích mé diplomové práce. Dále bych chtěla poděkovat své rodině za podporu během celého studia a všem ostatním lidem, kteří mi během studia pomáhali a podporovali mě.

Abstrakt

Tato práce se zabývá modelováním 1D přestupu tepla v konstrukci a posouzením konstrukce z hlediska požární odolnosti. Popisuje důležitá vstupní data pro modelování – např. nominální teplotní křivky, parametrickou křivku a vlastnosti materiálů. Také porovnává dostupné programy a ukazuje jednotlivé kódy z programování v prostředí Python.

V rámci této práce byl vytvořen kód, který pomocí metody konečných diferencí stanovuje rozložení teploty v železobetonové konstrukci a také kód, který pomocí proužkové metody zpřesňuje výpočty momentu únosnosti a posuzuje únosnost prvku za požární situace.

Klíčová slova

Sdílení tepla; Prostorová diskretizace; Časová diskretizace; Teplotní křivky ;
Parametrická křivka; Materiálové vlastnosti; Železobeton; Metoda konečných prvků; Python;
Proužková metoda; Metoda izotermy 500 °C; Zónová metoda

Abstract

This work deals with the modelling of 1D heat transport in structure and assessment of the structure in terms of fire resistance. It describes important input data for modelling such as: temperature curves, parametric curve and material properties. It also compares available programs and shows the individual codes at modelling in Python.

Within this work, a code was created that uses the finite difference method to determine the temperature distribution in a reinforced concrete structure, as well as a code that uses the strip method performing the calculations of load-bearing moments and evaluate the load-bearing capacity of the element in a fire situation.

Keywords

Heat transfer; Spatial discretization; Time discretization; Temperature curves; Parametric curve; Material properties; Reinforced Concrete; Finite difference methods; Python; Strip method; 500 °C Isotherm method; Zone method

Obsah

Zadání diplomové práce.....	II
Čestné prohlášení.....	III
Poděkování.....	IV
Abstrakt	V
Klíčová slova	V
Abstract	V
Keywords	V
Seznam použitých symbolů a zkratk.....	VIII

Úvod	1
Motivace	1
Cíle práce	3

1 Teoretická část.....	4
1.1 Teplotní křivky.....	5
1.1.1 Nominální teplotní křivky	5
1.1.2 Parametrická teplotní křivka	7
1.2 Teorie sdílení tepla.....	11
1.2.1 Matematický model.....	11
1.2.2 Diferenciální rovnice.....	12
1.3 Materiálové charakteristiky betonu za požáru	14
1.3.1 Měrné teplo	14
1.3.2 Objemová hmotnost betonu	15
1.3.3 Součinitel tepelné vodivosti + derivace podle teploty:	15
1.4 Numerické metody – Metoda konečných prvků	16
1.5 Numerické metody – Metoda konečných diferencí	17
1.5.1 Metoda konečných diferencí obecně.....	17
1.5.2 Metoda konečných diferencí pro vedení tepla	18
1.5.3 Časová a prostorová diskretizace	19
1.6 Numerická implementace modelu sdílení tepla	20
1.6.1 Rovnice pro výpočet teplot v prostředních buňkách.....	21
1.6.2 Rovnice pro výpočet teplot v krajních buňkách.....	21
1.7 Proužková metoda.....	23
1.8 Výpočetní metody pro posouzení požární odolnosti ŽB desky	26
1.8.1 Metoda izotermy 500 °C	26
1.8.2 Zónová metoda.....	32
1.9 Dostupné programy	37
1.9.1 TeDisOn.....	37
1.9.2 RCC _{fi}	38
1.9.3 FiDeS	39

1.9.4	ANSYS.....	41
1.10	Programovací jazyk.....	42
1.10.1	Matlab	42
1.10.2	Python	42
2	Praktická část – 1. blok	43
2.1	Teplotní křivky.....	44
2.2	Materiálové vlastnosti betonu	45
2.2.1	Měrné teplo	45
2.2.2	Objemová hmotnost	46
2.2.3	Součinitel tepelné vodivosti	47
2.2.4	Derivace součinitele tepelné vodivosti:	48
2.3	Modelování 1D sdílení tepla pro betonové konstrukce	49
2.3.1	Používané příkazy	49
2.3.2	Kód pro sdílení tepla	50
2.3.3	Výstupy z kódu	56
2.4	Porovnání výstupů.....	57
2.4.1	Požár z jedné strany	58
2.4.2	Požár ze dvou stran	60
2.4.3	Závěr	61
3	Praktická část – 2. blok	62
3.1	Vstupní údaje	63
3.1.1	Pracovní diagram betonu za běžné teploty	63
3.1.2	Pracovní diagram betonu za požární situace	65
3.1.3	Pracovní diagram oceli za běžné teploty.....	69
3.1.4	Pracovní diagram oceli za požáru	70
3.2	Modelování proužkové metody za běžné teploty	74
3.3	Porovnání výsledků (výpočet za běžné teploty)	75
3.4	Modelování proužkové metody za požáru	77
3.5	Porovnání výsledků (výpočet za požární situace).....	82
4	Praktická část – 3. blok	85
4.1	Izoterma 500 °C	86
4.2	Zónová metoda.....	89
4.2.1	Porovnání podrobnosti výpočtu	92
4.3	Porovnání výstupů.....	93
5	Shrnutí.....	95
6	Závěr	96
	Seznam obrázků	97
	Seznam tabulek.....	101
	Literatura	102
	Programy	103

Seznam použitých symbolů a zkratek

Latinské symboly

a_z, a_{500}	Tloušťka vyloučené vrstvy	m
A	Plocha	m^2
b	Šířka průřezu	m
c	Měrná tepelná kapacita	J/(kg·K)
d	účinná výška průřezu	m
D_t	Tepelná difuzivita	m^2/s
f_c	Pevnost betonu	Pa
f_y	Mez kluzu výztuže	Pa
F	Síla	N
h	Výška oken	m
k_c	Redukční součinitel pevnosti betonu	-
k_s	Redukční součinitel pevnosti výztuže	-
m	faktor spalování	-
M	Moment	Nm
O	Faktor otvorů	$m^{1/2}$
q	Hustota tepelného toku	W/m^2
t	Čas	min

Řecké symboly

α	Součinitel přestupu tepla	$W/(m^2 \cdot K)$
ε	Emisivita	-
η_{fi}	Redukční součinitel	-
Θ	Teplota	$^{\circ}C$
λ	Součinitel tepelné vodivosti	$W/(m \cdot K)$
ρ	Objemová hmotnost	kg/m^3
σ	Stefan-Boltzmannova konstanta	$W/(m^2 \cdot K^4)$
ϕ	Polohový faktor	-

Zkratky

IZO500	Metoda izotermy 500 °C
PÚ	Požární úsek
ČVUT	České vysoké učení technické
ŽB	Železobeton

Úvod

Diplomová práce je zaměřena na sestavení kódu pro posouzení požární odolnosti železobetonové desky, který bude obsahovat výpočet 1D prostupu tepla konstrukcí a jednotlivé části posouzení. Vše v programovacím prostředí Python. Tato práce tedy bude hlavně o vstupech do programu, tedy o použitých rovnicích, materiálech atd. a také o výstupech, které budou průběžně získávány při tvorbě programu.

Motivace

Motivací bylo navázání na předmět 134NKPZ-Nosné konstrukce za požáru, kde jsem si v excelu naprogramovala výpočet prostupu tepla betonovou konstrukcí a chtěla jsem v tomto tématu pokračovat a naprogramovat kód, který umí spočítat prostup tepla v železobetonové desce a bude zároveň umět posoudit požární odolnost.

Ukázky z excelu, na který budu navazovat vytvořením programu (Obr. 1 až Obr. 4):

Numerická analýza transportu tepla v betonových konstrukcích		metoda konečných diferencí	
Počáteční podmínka:			
pro $x \in (0;L)$			
$t = 0$	$\Theta(x,0) = \Theta_0$	=> konstantní teplota = teplota okolí	
Okrajová podmínka:			
pro $x=0$ nebo $x=L$			
$t \in (0;\tau)$			
-> normová teplotní podmínka			
		$-\lambda \cdot \frac{\partial \theta}{\partial x} + n = \alpha_c \cdot (\theta - \theta_\infty) + \Sigma \sigma \cdot ((\theta + 237,15)^4 - (\theta_\infty + 237,15)^4)$	
-> vystavený povrch:			
$\Theta(\infty) =$	345 $\log_{10}(8t+1)+20$ °C	teplota plynů - normová křivka (může být libovolná) (t v min.)	
$d(c) =$	25 $W/(m^2 \cdot K)$	normová křivka = 25 (parametrická = 35, uhlodíková = 50)	
$\epsilon =$	0,7	emisivita povrchu - pro beton (jinak dle ČSN EN 199x-1-2)	
$\sigma =$	5,67E-08 $W/(m^2 \cdot K^4)$	Stefan-Boltzmanova konstanta	
-> nevystavený povrch:			
$\Theta(\infty) =$	20,00 °C	pro beton (jinak dle ČSN EN 199x-1-2)	
$d(c) =$	4 $W/(m^2 \cdot K)$	přesnější (tze $\alpha = 9$, $\epsilon = 0$)	
$\epsilon =$	0,7		
Počáteční teplota	$\Theta =$	20,00 °C	
Rozměr kce:	$L =$	0,25 m	
	$\Delta x =$	0,0125 m	pro 20 dílků
Hustota při 20°C:	$\rho =$	2350 kg/m^3	
Měrná tepelná kapacita:	$c =$	dle teploty $J/kg \cdot K$	
	$\Delta t =$	10 s	
	převod K	273,15	
	max. časový krok:	84,67 s	$\Delta t_{crit} = \frac{(\Delta x)^2}{2 \cdot \lambda / \rho \cdot c}$

Obr. 1 - Ukázka zadávání vstupních údajů

Pomocné vzorečky:

dle normy:

$$\lambda_{(\theta)} = 2 - 0,2451 * \frac{\theta}{100} + 0,0107 * \left(\frac{\theta}{100}\right)^2 = 2 - 2,451 * 10^{-3} * \theta + 1,07 * 10^{-6} * \theta^2$$

$$\frac{\partial \lambda}{\partial \theta} = -\frac{0,2451}{100} + 2 * \frac{0,0107}{100^2} * \theta = -2,451 * 10^{-3} + 2,14 * 10^{-6} * \theta$$

$$\frac{\partial \theta}{\partial x} = \frac{\theta_{i+1} - \theta_{i-1}}{2 * \Delta x}$$

$$\frac{\partial^2 \theta}{\partial x^2} = \frac{\theta_{i+1} - 2 * \theta_i + \theta_{i-1}}{\Delta x^2}$$

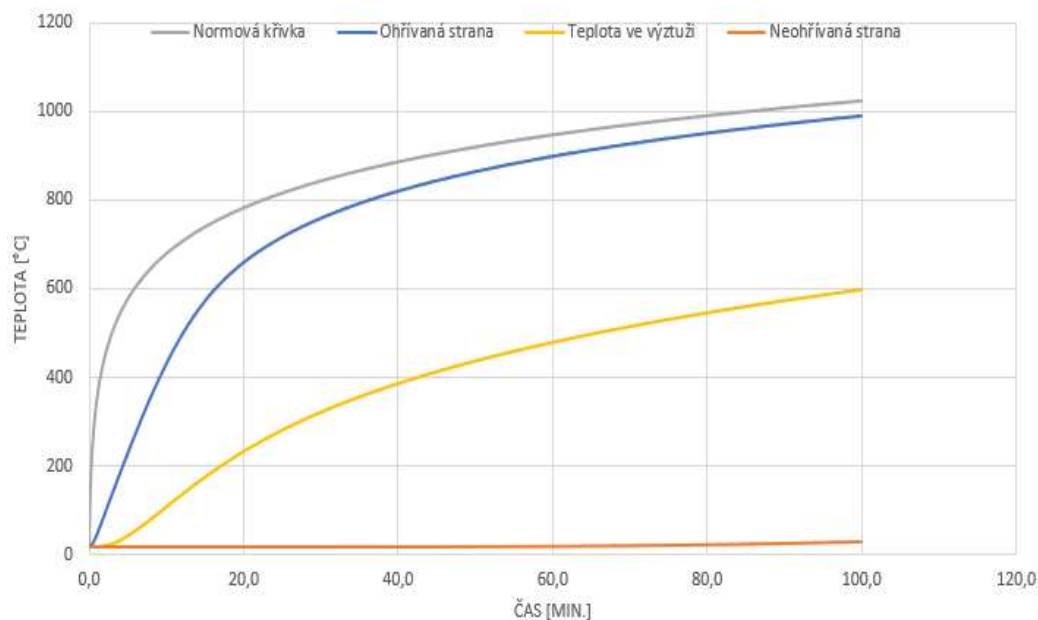
$$\frac{\partial \theta}{\partial t} = \frac{\theta_{k+1} - \theta_{k-1}}{\Delta t}$$

Obr. 2 - Ukázka použitých vzorečků

q (pomocně-vystavený povrch)			k (s) \ i (m)			
	hod	min	0	1	2	3
			0	0,0125	0,025	0,0375
	0,00	0,000	0	20	20	20
	0,00	0,167	10	20,0	20,0	20,0
	-83,16	0,333	20	20,1	20,0	20,0
	-143,38	0,500	30	20,2	20,0	20,0
	-206,95	0,667	40	20,3	20,0	20,0
	-277,19	0,833	50	20,5	20,0	20,0

Obr. 3 - Ukázka výpočetní tabulky

ROZLOŽENÍ TEPLoty V ČASE V BETONOVÉ STĚNĚ



Obr. 4 - Výstup - rozložení teploty v čase

Cíle práce

Hlavními cíli práce jsou:

- 1) Popis matematického modelu sdílení tepla
- 2) Popis metody prostorové a časové diskretizace
- 3) Popis numerické metody a okrajových podmínek
- 4) Popis vybraných metod (Proužková metoda, Metoda izotermy 500 °C, Zónová metoda)
- 5) Počítačová implementace
- 6) Aplikace na konkrétních příkladech

Kapitola 1

1 Teoretická část

Teoretickou část jsem rozdělila na devět bloků, přičemž každý se věnuje popisu jiné problematiky.

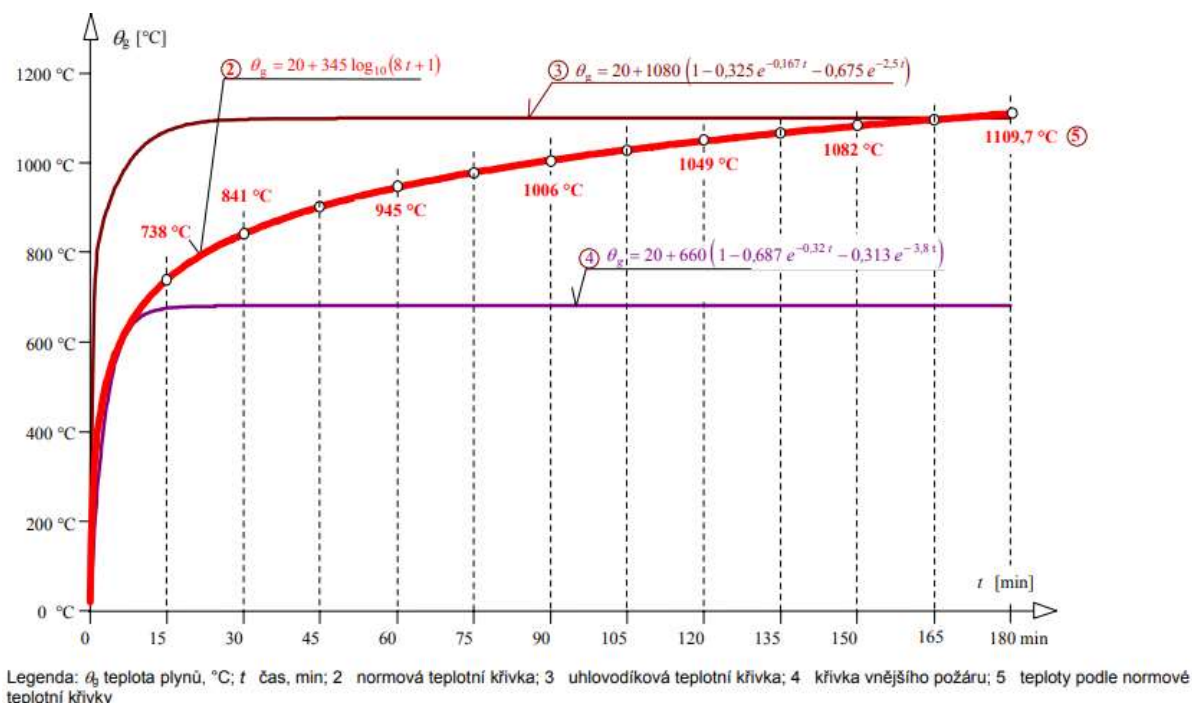
1. První blok se věnuje popisu jednotlivých teplotních křivek.
2. Druhý blok se věnuje teorii sdílení tepla. Zde popisují matematický model popsaný diferenciálními rovnicemi.
3. Třetí blok se věnuje materiálovým charakteristikám betonu za požáru.
4. Čtvrtý blok se věnuje popisu numerické metody konečných prvků.
5. Pátý blok se věnuje popisu numerické metody konečných diferencí.
6. Šestý blok se věnuje numerické implementaci modelu sdílení tepla.
7. Sedmý blok se věnuje popisu proužkové metody.
8. Osmý blok se věnuje výpočetním metodám pro posouzení požární odolnosti. Zde popisují metodu izotermy 500 °C a zónovou metodu.
9. Devátý blok se věnuje dostupným programům. Zde jsem popsala programy TeDisOn [1P], RCCfi [2P], FiDeS [3P] a ANSYS [4P].
10. Desátý blok se věnuje popisu programovacího jazyka. Popsala jsem zde programovací jazyk Matlab a Python.

1.1 Teplotní křivky

Podle eurokódu EN 1991-1-2 [1] existují dva typy křivek, a to nominální teplotní křivky a parametrická křivka. Největší rozdíl je v tom, že nominální křivky uvažují jen fázi ohřevu a parametrická křivka počítá i s fází chladnutí. Dále se také liší v zadávání parametrů, kdy u nominálních křivek pro výpočet stačí pouze čas a u parametrické křivky se zadávají specifikace požárního úseku (např. podlahová plocha, otvory, materiálové charakteristiky konstrukcí, ...)

1.1.1 Nominální teplotní křivky

V normě EN 1991-1-2 [1] jsou popsány celkem tři nominální teplotní křivky, a to normová teplotní křivka, křivka vnějšího požáru a uhlovodíková (Obr. 5). Křivky udávají teplotu plynů v požárním úseku v závislosti na čase. Tyto křivky se liší vývojem tepla, použitím a maximální dosaženou teplotou [1].



Obr. 5 - Nominální teplotní křivky[2]

1.1.1.1 Normová (ISO 834)

Normová teplotní křivka je nejpoužívanější nominální křivka a vznikla na základě statistiky, kdy byl mezinárodně stanoven jednotný časový průběh teploty při požáru. Tato křivka má rostoucí průběh, kdy na začátku je prudký rozvoj teploty a s přibývajícím časem postupně zpomaluje.

Předpis pro teplotu:

$$\theta_g = 20 + 345 \cdot \log_{10}(8 \cdot t + 1) \quad (1)$$

- θ_g = teplota plynu v požárním úseku [$^{\circ}\text{C}$]
- t = čas [min]

Pro normovou teplotní křivku je součinitel přestupu tepla: $\alpha_c = 25 \text{ [W / m}^2 \cdot \text{K]}$

1.1.1.2 Vnějšího požáru

Tato křivka, jak je již z názvu patrné, se využívá pro simulaci vnějšího požáru. Uvažuje tedy dostatečné množství kyslíku a také nižší teploty než při požáru uvnitř budovy.

Teplota plynů je limitována hodnotou $680 \text{ }^{\circ}\text{C}$, přes kterou nikdy nepřestoupí.

Předpis pro teplotu:

$$\theta_g = 660 \cdot (1 - 0,687 \cdot e^{-0,32t} - 0,313 \cdot e^{-3,8t}) + 20 \quad (2)$$

- θ_g = Teplota plynu [$^{\circ}\text{C}$]
- t = čas [min]

Pro křivku vnějšího požáru je součinitel přestupu tepla: $\alpha_c = 25 \text{ [W / m}^2 \cdot \text{K]}$

1.1.1.3 Uhlovodíková

Uhlovodíková křivka se používá především při hoření ropných produktů (např. garáže), kdy se uvažuje rychlejší nárůst teploty než při klasickém požáru.

Teplota plynů je zde také limitována, ale hodnotou vyšší, a to $1100 \text{ }^{\circ}\text{C}$.

Předpis pro teplotu:

$$\theta_g = 1080 \cdot (1 - 0,325 \cdot e^{-0,167t} - 0,675 \cdot e^{-2,5t}) + 20 \quad (3)$$

- θ_g = Teplota plynu v požárním úseku [$^{\circ}\text{C}$]
- t = čas [min]

Pro uhlovodíkovou teplotní křivku je součinitel přestupu tepla: $\alpha_c = 50 \text{ [W / m}^2 \cdot \text{K]}$

1.1.2 Parametrická teplotní křivka

Parametrická křivka se používá pro reálnější vykreslení průběhu teplot. Je složena ze dvou částí, a to z fáze ohřevu, kdy teplota plynů narůstá až do maximální teploty (viz kapitola 1.1.2.2) a z fáze chladnutí, kdy nastává pokles teploty až na 20°C.

Parametrická křivka může být dvojího typu – řízená palivem (kapitola 1.1.2.4) nebo řízena ventilací (kapitola 1.1.2.3), záleží, co je zde rozhodující faktor, zda dřív dojde palivo nebo kyslík. Podle toho se určí i vzorečky pro výpočet.

Tato křivka se smí použít jen při splnění stanovených podmínek:

- Požární úsek do podlahové plochy 500 m², bez otvorů ve střeše a s maximální výškou požárního úseku 4 m.

Předpokládá se zde úplné vyhoření požárního zatížení[1].

1.1.2.1 Teplotní křivka ve fázi ohřevu

První fází je fáze ohřevu, která nastává od začátku hoření a končí, když v požárním úseku dojde kyslík nebo palivo. Následně nastává přelom (Maximální teplota) a fáze chladnutí.

Předpis pro fázi ohřevu:

$$\theta_g = 20 + 1\,325 \cdot (1 - 0,324 \cdot e^{-0,2 \cdot t^*} - 0,204 \cdot e^{-1,7 \cdot t^*} - 0,472 \cdot e^{-19 \cdot t^*}) \quad (4)$$

- θ_g = teplota plynů v požárním úseku [°C]

$$t^* = t \cdot \Gamma [h] \quad (5)$$

- t = čas [h]

$$\Gamma = \left(\frac{O}{b}\right)^2 / \left(\frac{0,04}{1\,160}\right)^2 [-] \quad (6)$$

$$b = \sqrt{\rho \cdot c \cdot \lambda} [J/m^2 \cdot s^{1/2} \cdot K] \quad (7)$$

$$100 \leq b \leq 2\,200 \quad (8)$$

- ρ = objemová hmotnost konstrukcí ohraničujících požární úsek [kg/m³]
- c = specifické teplo konstrukcí ohraničujících požární úsek [J/kg * K]
- λ = tepelná vodivost konstrukcí ohraničujících požární úsek [W/m * K]
- O = faktor otvorů [m^{1/2}]

$$O = (A_v \cdot \sqrt{h_{eq}}) / A_t \quad (9)$$

$$0,02 \leq O \leq 0,20 \quad (10)$$

- A_v = celková plocha svislých otvorů ve všech stěnách ohraničujících požární úsek [m^2]
- h_{eq} = vážený průměr výšek oken ve všech stěnách ohraničujících požární úsek [m]
- A_t = celková plocha konstrukcí ohraničujících požární úsek [m^2]

1.1.2.2 Maximální teplota

Maximální teplota ve fázi ohřevu nastává v čase t^*_{max} . V tomto čase přechází křivka z rostoucí fáze do klesající části.

$$t^*_{max} = t_{max} \cdot \Gamma \quad (11)$$

$$t_{max} = \max\{t_{lim,pal}; t_{lim,vent}\} \quad (12)$$

- $q_{t,d}$ = návrhová hodnota hustoty požárního zatížení, vztažená k celé ploše povrchu A_t

$$q_{t,d} = q_{f,d} \cdot A_f / A_t \quad (13)$$

$$50 \leq q_{t,d} \leq 1\,000 \text{ [MJ/m}^2\text{]} \quad (14)$$

- $q_{f,d}$ = návrhová hodnota hustoty požárního zatížení, vztažená k ploše podlahy A_f

$$q_{f,d} = q_{f,k} \cdot m \cdot \delta_{q1} \cdot \delta_{q2} \cdot \delta_n \quad (15)$$

- m = faktor spalování
- δ_{q1} = faktor zohledňující riziko požáru vzhledem k velikosti prostoru
- δ_{q2} = faktor zohledňující riziko požáru vzhledem k typu obsazenosti
- δ_n = faktor zohledňující různá aktivní protipožární opatření
- $q_{f,k}$ = charakteristická hodnota hustoty požárního zatížení, vztažená k ploše podlahy
- $t_{lim,pal}$ = časová hodnota rychlosti požáru pro požár řízený palivem
 - o 25 min -> malá rychlost rozvoje požáru
 - o 20 min -> střední rychlost rozvoje požáru
 - o 15 min -> velká rychlost rozvoje požáru
- $t_{lim,vent}$ = časová hodnota rychlosti požáru pro požár řízený ventilací
 - o $t_{lim,vent} = (0,2 \cdot 10^{-3} \cdot q_{t,d})$

1.1.2.3 Požár řízený ventilací

Pokud vyjde $t_{ma} = t_{lim,vent}$, pak je požár řízený ventilací a výpočet probíhá podle vzorečků uvedených výše (4) - (9).

1.1.2.4 Požár řízený palivem

Pokud vyjde $t_{max} = t_{lim,pal}$, pak je požár řízený palivem a budou se muset upravit vzorečky, a to tak, že čas t^* používaný ve vzorečku (4) se nahradí takto:

$$t^* = t \cdot \Gamma_{lim} [h] \quad (16)$$

- Γ_{lim} :

$$\Gamma_{lim} = \left(\frac{O_{lim}}{b}\right)^2 / \left(\frac{0,04}{1160}\right)^2 [-] \quad (17)$$

- Faktor otvorů:

$$O_{lim} = 0,1 \cdot 10^{-3} \cdot q_{t,d} / t_{lim,pal} \quad (18)$$

- Pokud je $O > 0,04$ (9) a (13) a $b < 1160$ (7), pak se Γ_{lim} (17) vynásobí „k“

$$k = 1 + \left(\frac{O - 0,04}{0,04}\right) \cdot \left(\frac{q_{t,d} - 75}{75}\right) \cdot \left(\frac{1160 - b}{1160}\right) \quad (19)$$

o Tedy

$$t^* = k \cdot t \cdot \Gamma_{lim} [h] \quad (20)$$

1.1.2.5 Teplotní křivka ve fázi chladnutí

Fáze chladnutí nastává hned po maximální teplotě, tedy od času t^*_{max} po takový čas, ve kterém se teplota dostane na počátečních 20 °C. Pak už se předpokládá, že požár dohořel a od tohoto času dál je jen konstantních 20 °C.

Parametr x vstupující do výpočtu se určí následujícím způsobem:

- Pokud je požár řízený palivem, tedy $t_{max} = t_{lim,pal}$, pak platí:

$$x = t_{lim,pal} \cdot \Gamma / t^*_{max} \quad (21)$$

- Pokud je požár řízený ventilací, tedy $t_{max} = t_{lim,vent}$, pak platí:

$$x = 1 \quad (22)$$

Vztahy pro fázi chladnutí jsou dány podle času t^*_{max} . Čas t^* je dán dle rovnice (5) a koeficient x je dán dle rovnic (21) a (22)

$$t^*_{max} = t_{lim,vent} \cdot \Gamma \quad (23)$$

- pro: $t^*_{max} \leq 0,5$

$$\theta_g = \theta_{max} - 625 \cdot (t^* - t^*_{max} \cdot x) \quad (24)$$

- pro: $0,5 < t^*_{max} < 2$

$$\theta_g = \theta_{max} - 250 \cdot (3 - t^*_{max})(t^* - t^*_{max} \cdot x) \quad (25)$$

- pro: $2 \leq t^*_{max}$

$$\theta_g = \theta_{max} - 250 \cdot (t^* - t^*_{max} \cdot x) \quad (26)$$

1.2 Teorie sdílení tepla

Sdílení tepla je jedním ze způsobů přenosu energie. Spočívá v tepelné výměně, která probíhá tak, že teplejší těleso předává část své vnitřní energie chladnějšimu tělesu. Výměna tepla může probíhat vedením (kondukcí), sáláním (radiací) nebo prouděním (konvekcí).

V této práci se budu věnovat hlavně přestupu tepla vedením neboli kondukcí, které probíhá uvnitř konstrukce.

Při vedení tepla částice látky v oblasti s vyšší teplotou předávají část své energie prostřednictvím vzájemných srážek částicím s nižší teplotou (majícím nižší energii). Při tomto procesu se částice nepřemísťují, ale kmitají kolem svých rovnovážných poloh [3].

Vedení tepla lze také rozdělit na ustálené a neustálené vedení. Ustálené neboli stacionární vedení tepla není závislé na čase, což znamená, že teplotní rozdíl mezi částmi těles je stálý. Při neustáleném neboli nestacionárním vedení tepla dochází k postupnému vyrovnávání teplotních rozdílů mezi jednotlivými částmi tělesa [3].

Základním zákonem vedení tepla je Fourierův zákon, který je matematicky popsán níže. Tento zákon vyplývá z experimentálně zjištěných skutečností [3].

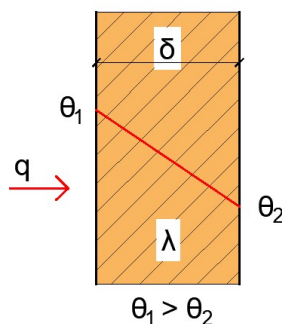
1.2.1 Matematický model

První Fourierův zákon udává teplotní spád, neboli hustotu tepelného toku q , vyjádřenou součinem gradientu teploty θ [K/m] a tepelné vodivosti λ [W/mK].

$$q_{(x)} = -\lambda_{(x)} \cdot \text{grad } \theta_{(x)} \quad (27) [4]$$

Úpravou Fourierova zákona se získá vztah pro plošnou hustotu tepelného toku jednoduchou rovinnou konstrukcí: [4]

$$q = -\lambda \cdot \frac{d\theta}{dx} = \frac{(\theta_1 - \theta_2)}{\frac{\delta}{\lambda}} \quad (28)$$



Obr. 6 - Jednosměrné vedení tepla (převzato z [4])

Tepelný tok je množství tepla, které projde jednotkovou plochou $A [1m^2]$ s normálou n , za jednotku času $t [s]$. Tepelný tok lze rozdělit do dvou směrů, v tomto případě se ale budu zabývat pouze tepelným tokem jednosměrným (tedy v 1D rovině) viz Obr. 6.

1.2.2 Diferenciální rovnice

Tato diferenciální rovnice popisuje 1D transport tepla za požáru:

$$\rho \cdot c_p \cdot \frac{\partial \theta}{\partial t} - \frac{\partial}{\partial x} \cdot \left(\lambda \cdot \frac{\partial \theta}{\partial x} \right) = 0 \quad (29)$$

1.2.2.1 Počáteční podmínka

Pro celý průřez, tedy pro $x \in (0; L)$, v čase $t = 0$, bude konstantní teplota, a to teplota okolí $\theta_{(x,0)} = \theta_{(0)}$, většinou 20 °C.

1.2.2.2 Okrajové podmínky

Pro $x = 0$ nebo $x = L$; $t \in (0; \tau)$

1. Dirichletova okrajová podmínka

Je zde předepsaná teplota na hranici (např. naměřená termočlánky)

$$\theta_{(0,t)} = \theta_{(L,t)} = \bar{\theta}_{(t)} \quad (30)$$

2. Neumannova okrajová podmínka

Je zde předepsán tepelný tok na hranici

- Pro homogenní materiál je tepelný tok na hranici 0 (hranice je dokonale tepelně izolovaná)

$$q \cdot n_{(x)} = 0 \quad (31)$$

- Tepelný tok:

$$q = -\lambda_x \cdot \frac{\partial \theta_x}{\partial x} \quad (32)$$

3. Cauchyho přestup tepla na hranici

Zde je předepsaný tepelný tok z teploty okolí

$$q \cdot n_{(x)} = \bar{q} = \alpha_{c(x)} (\theta_{(x)} - \theta_{\infty(x)}) \quad (33)$$

4. Tepelný tok od radiace

Zde je předepsaný tepelný tok od radiace, přičemž teploty se uvažují v Kelvinech.

$$q \cdot n_{(x)} = \varepsilon_{(x)} \cdot \sigma_{(x)} \cdot (\theta_{(x)}^4 - \theta_{\infty(x)}^4) \quad (34)$$

5. Normová okrajová podmínka

Normová okrajová podmínka je kombinací všech předchozích podmínek. Pokud bude $\alpha_{c(x)} = 0$, pak dostaneme 4. podmínku, pokud bude $\varepsilon_{(x)} = 0$, pak dostaneme 3. podmínku, když $\varepsilon_{(x)} = 0; \alpha_{c(x)} = 0$, pak dostaneme Neumannovu podmínku a pokud $\varepsilon_{(x)} = 0; \alpha_{c(x)} = \infty$, pak dostaneme Dirichletovu podmínku.

$$q \cdot n_{(x)} = \alpha_{c(x)}(\theta_{(x)} - \theta_{\infty(x)}) + \varepsilon_{(x)} \cdot \sigma_{(x)} \cdot (\theta_{(x)}^4 - \theta_{\infty(x)}^4) \quad (35)$$

Prvky vstupující do normové okrajové podmínky:

- Na vystaveném povrchu:
 - Teplota plynů - $\theta_{\infty(x)} = \theta_{g(x)}$; platí pro libovolnou křivku
 - Součinitel přestupu tepla [1] $\alpha_{c(x)} = 25 \text{ W/m}^2\text{K}$; platí pro normovou křivku
 $\alpha_{c(x)} = 25 \text{ W/m}^2\text{K}$; platí pro křivku vnějšího požáru
 $\alpha_{c(x)} = 35 \text{ W/m}^2\text{K}$; platí pro parametrickou křivku
 $\alpha_{c(x)} = 50 \text{ W/m}^2\text{K}$; platí pro uhlovodíkovou křivku
 - Emisivita - $\varepsilon = \varepsilon_m \cdot \Phi \cdot \varepsilon_f$
 - Emisivita požáru - $\varepsilon_f = 1$
 - Polohový faktor - $\Phi = 1$; strana bezpečnosti
 - Emisivita materiálu - $\varepsilon_m = 0,7$; Betonové konstrukce (EN 1991-1-2 [1])
 - Stefan-Boltzmannova konstanta - $\sigma = 5,67 \cdot 10^{-8} \text{ W/m}^2\text{K}^4$
- Nevystavený povrch:
 - Přesnější výpočet: $\alpha_{c(x)} = 4 \text{ W/m}^2\text{K}$ $\varepsilon_m = 0,7$ (dle materiálu)
 - Strana bezpečnosti: $\alpha_{c(x)} = 0 \text{ W/m}^2\text{K}$ $\varepsilon_m = 0,0$ -> zanedbání ochlazování
 $\alpha_{c(x)} = 9 \text{ W/m}^2\text{K}$ $\varepsilon_m = 0$

1.3 Materiálové charakteristiky betonu za požáru

Veškeré materiálové charakteristiky za požáru byly brány z příslušného eurokódu (ČSN EN 1992-1-2 [5])

Dále uvedu vzorečky pro jednotlivé materiálové charakteristiky betonu za požáru, sloužící jako vstupní údaje ke tvorbě programu.

1.3.1 Měrné teplo

Beton – suchý (0% vlhkost)

- V rozmezí teplot 20 – 100 °C

$$c_p(\theta) = 900 \text{ [J / kgK]} \quad (36)$$

- V rozmezí teplot 100 – 200 °C

$$c_p(\theta) = 900 + (\theta - 100) \quad (37)$$

- V rozmezí teplot 200 – 400 °C

$$c_p(\theta) = 1\,000 + (\theta - 200)/2 \quad (38)$$

- V rozmezí teplot 400 – 1 200 °C

$$c_p(\theta) = 1\,100 \text{ [J / kgK]} \quad (39)$$

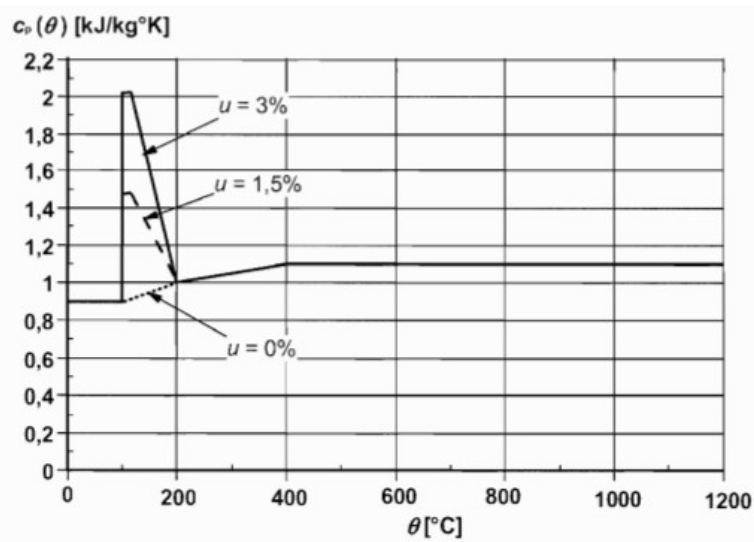
Beton – s vlhkostí: (pro tento beton platí, že mezi 100 – 115°C mají níže vypsané hodnoty, při 200 °C mají 1 000 J/kgK a pak už pokračují stejně jako suchý beton – viz Obr. 7)

- 1,5 %

$$c_p(\theta) = 1\,470 \text{ [J / kgK]} \quad (40)$$

- 3,0 %

$$c_p(\theta) = 2\,020 \text{ [J / kgK]} \quad (41)$$



Obr. 7 - Měrné teplo v závislosti na teplotě pro jednotlivé vlhkosti [5]

1.3.2 Objemová hmotnost betonu

- Pro teploty 20 – 115 °C

$$\rho(\theta) = \rho(20^{\circ}\text{C}) \quad (42)$$

- Pro teploty 115 – 200 °C

$$\rho(\theta) = \rho(20^{\circ}\text{C}) \cdot (1 - 0,02 \cdot (\theta - 115) / 85) \quad (43)$$

- Pro teploty 200 - 400 °C

$$\rho(\theta) = \rho(20^{\circ}\text{C}) \cdot (0,98 - 0,03 \cdot (\theta - 200) / 200) \quad (44)$$

- Pro teploty 400 – 1 200 °C

$$\rho(\theta) = \rho(20^{\circ}\text{C}) \cdot (0,95 - 0,07 \cdot (\theta - 400) / 800) \quad (45)$$

1.3.3 Součinitel tepelné vodivosti + derivace podle teploty:

Zde jsou uvedeny součinitelé tepelné vodivosti a jejich derivace, které budou použité ve výpočtu přestupu tepla (Obr. 8). (Pro beton platí v rozmezí 20 – 1 200 °C.)

- Horní mez tepelné vodivosti:

$$\lambda_c = 2 - 0,2451 \cdot (\theta/100) + 0,0107 \cdot (\theta/100)^2 \text{ [W/mK]} \quad (46)$$

Derivace λ_c podle teploty:

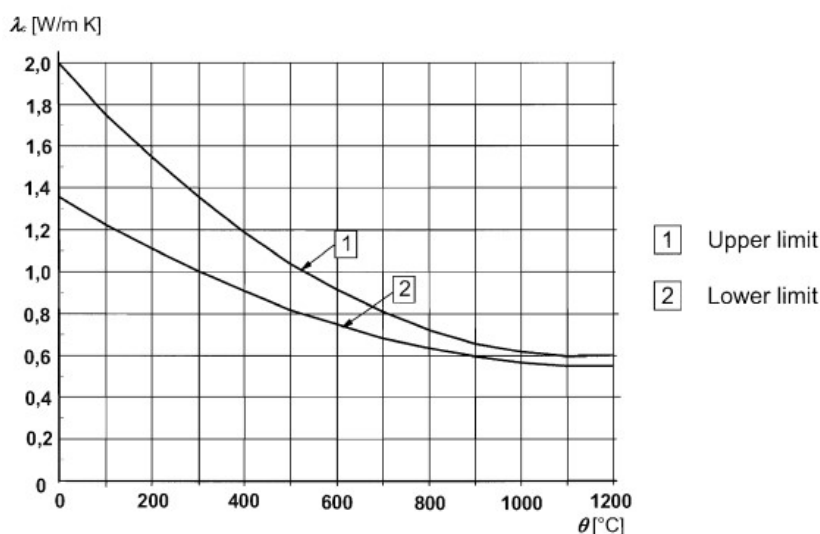
$$\frac{\partial \lambda}{\partial \theta} = -\frac{0,2451}{100} + 2 \cdot \frac{0,0107 \cdot \theta}{100^2} = -2,451 \cdot 10^{-3} + 2,14 \cdot 10^{-6} \cdot \theta \quad (47)$$

– Spodní mez tepelné vodivosti:

$$\lambda_c = 1,36 - 0,136 \cdot (\theta/100) + 0,0057 \cdot (\theta/100)^2 \quad [W/mK] \quad (48)$$

Derivace λ_c podle teploty:

$$\frac{\partial \lambda}{\partial \theta} = -\frac{0,136}{100} + 2 \cdot \frac{0,0057 \cdot \theta}{100^2} = -1,36 \cdot 10^{-3} + 1,14 \cdot 10^{-6} \cdot \theta \quad (49)$$



Obr. 8 - Změna horní (Upper) a spodní (Lower) meze tepelné vodivosti s teplotou [5]

1.4 Numerické metody – Metoda konečných prvků

Metoda konečných prvků anglicky nazývaná Finite Element Method (FEM) je nejčastěji užívaná systematická a univerzální metoda pro numerické řešení problémů.

Metoda je založena na principu virtuálních prací, řeší podmínky rovnováhy vnějších a vnitřních sil [6].

Touto metodou se rozdělí řešené oblasti na menší části, tzv. konečné prvky nebo také elementy, jejichž chování lze jednoduše popsat – obvykle použitím maticového zápisu. Dalším krokem je složení jednotlivých elementů zpět do celkové konstrukce [7].

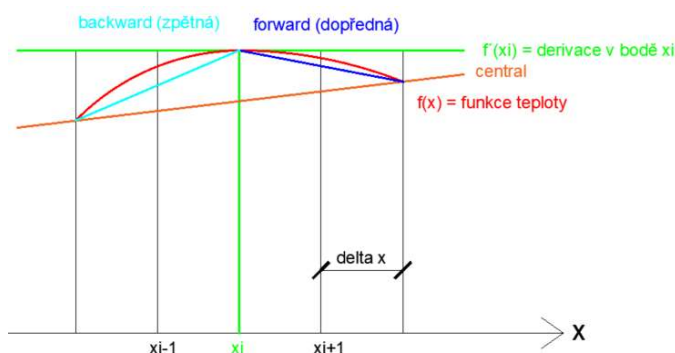
1.5 Numerické metody – Metoda konečných diferencí

Metoda konečných diferencí anglicky nazývaná Finite Difference Method (FDM), v češtině také známá jako Metoda sítí.

Tato metoda je založena na diskretizaci parciálních diferenciálních rovnic popisujících daný problém. Rovnice jsou diskretizovány v prostoru i v čase.

1.5.1 Metoda konečných diferencí obecně

Zde uvedu postup náhrady derivací diferencemi (Obr. 9).



Obr. 9 - Náhrada derivací diferencemi (Vytvořeno v AutoCADu)

Jednotlivé kroky Δx volím co nejmenší, pro přesnější řešení. Pro jednotlivé kroky přičítám nebo odečítám Δx , pro krok zpět: $x_{i-1} = x_i - \Delta x$; pro krok vpřed $x_{i+1} = x_i + \Delta x$.

Jednotlivé náhrady derivací diferencemi:

- Dopředná (forward) diference (D_+ vychází záporně, protože funkce $f(x)$ klesá)

$$f'_{(xi)} = \left(\frac{df}{dx} \right)_{xi} \approx \Rightarrow D_+ \quad (50)$$

- Zpětná (backward) diference

$$f'_{(xi)} = \left(\frac{df}{dx} \right)_{xi} \approx \frac{f_{(xi)} - f_{(xi-1)}}{\Delta x} \Rightarrow D_- \quad (51)$$

- Střední (central) diference

$$f'_{(xi)} = \left(\frac{df}{dx} \right)_{xi} \approx \frac{f_{(xi+1)} - f_{(xi-1)}}{2\Delta x} \Rightarrow D_0 \quad (52)$$

Náhrada druhé derivace diferencemi:

$$f''_{(xi)} = \left(\frac{d^2f}{dx^2} \right)_{xi} \approx \frac{D_+ - D_-}{\Delta x} = \frac{f_{(xi+1)} - 2f_{(xi)} + f_{(xi-1)}}{(\Delta x)^2} \quad (53)$$

1.5.2 Metoda konečných diferencí pro vedení tepla

Střední (central) diference

$$\frac{\partial \theta}{\partial x} = \frac{\theta_{(i+1)} - \theta_{(i-1)}}{2\Delta x} \quad (54)$$

Druhá derivace

$$\frac{\partial^2 \theta}{\partial x^2} = \frac{\theta_{(i+1)} - \theta_{(i)} + \theta_{(i-1)}}{(\Delta x)^2} \quad (55)$$

Dopředná (forward) diference

$$\frac{\partial \theta}{\partial t} = \frac{\theta_{(i+1)} - \theta_{(i)}}{\Delta t} \quad (56)$$

Sestavení rovnic pro vedení tepla

$$\rho_{(\theta)} \cdot C_{P(\theta)} \cdot \frac{\partial \theta}{\partial t} - \left(\frac{\partial \lambda}{\partial x} \cdot \frac{\partial \theta}{\partial x} + \lambda \cdot \frac{\partial^2 \theta}{\partial x^2} \right) = 0 \quad (57)$$

– Derivace λ podle x lze přepsat následovně:

$$\frac{\partial \lambda}{\partial x} = \frac{\partial \lambda}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} \quad (58)$$

– Po dosazení do rovnice vznikne:

$$\rho_{(\theta)} \cdot C_{P(\theta)} \cdot \frac{\partial \theta}{\partial t} - \left(\frac{\partial \lambda}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} \cdot \frac{\partial \theta}{\partial x} + \lambda \cdot \frac{\partial^2 \theta}{\partial x^2} \right) = 0 \quad (59)$$

– Konečná rovnice vedení tepla:

$$\rho_{(\theta)} \cdot C_{P(\theta)} \cdot \frac{\partial \theta}{\partial t} - \frac{\partial \lambda}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial x} \right)^2 - \lambda \cdot \frac{\partial^2 \theta}{\partial x^2} = 0 \quad (60)$$

– Pro konstantní součinitel vodivosti (λ) by pak rovnice vypadala následovně:

$$\rho_{(\theta)} \cdot C_{P(\theta)} \cdot \frac{\partial \theta}{\partial t} - \lambda \cdot \frac{\partial^2 \theta}{\partial x^2} = 0 \quad (61)$$

1.5.3 Časová a prostorová diskretizace

Jedná se o náhradu spojitého prostředí systémem diskrétních bodů. Pro lepší přehlednost zde uvedu obrázek i vzorce.

1.5.3.1 Časová diskretizace

Pro přesnější výpočet musíme počítat po malých časových krocích (v řádu sekund). Časový krok (Δt) lze určit nebo dopočítat – pokud je znám celkový čas (τ) a počet kroků (n)

$$\Delta t = \frac{\tau}{n} \quad (62)$$

Je zde ale omezení kvůli náhradě časové derivace. Aby výpočty fungovaly, musí být krok dostatečně malý. Zde je vzoreček pro výpočet maximálního časového kroku.

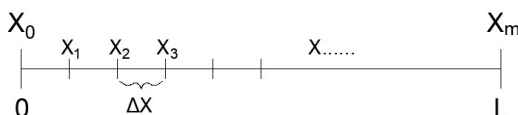
$$\Delta t_{crit} = \frac{(\Delta x)^2}{2 \cdot D_t} \quad (63)$$

kde D_t je tepelná difuzivita (teplotní vodivost), která vyjadřuje schopnost látky vyrovnávat rozdílné teploty při neustáleném šíření tepla vedením v homogenním prostředí.

$$D_t = \lambda / (\rho \cdot C_p) \quad (64)$$

1.5.3.2 Prostorová diskretizace

Při řešení 1D problému je třeba šířku konstrukce rozdělit na „m“ malých proužků, které následně budou zkoumány (viz Obr. 10). Čím více proužků, tím přesnější výpočet bude, ale může se stát, že se tím zpomalí výpočetní čas.



Obr. 10 - Prostorová diskretizace v 1D

Při dané šířce konstrukce si musím určit, zda konstrukci rozdělím na „m“ proužků nebo jak velké tyto proužky budou. Druhou veličinu si následně dopočítám.

Pokud znám počet proužků, dopočítám délku kroku.

$$\Delta x = \frac{L}{m} \quad (65)$$

Pokud znám délku kroku, dopočítám počet proužků.

$$m = \frac{L}{\Delta x} \quad (66)$$

1.6 Numerická implementace modelu sdílení tepla

Zde uvedu, co se bude modelovat a vysvětlím princip, jak funguje tento výpočet vedení tepla.

Na obrázku níže (Obr. 11) je vidět tabulka. Počet řádků je počet časových kroků (např. chci vědět teplotu v čase 100 s a časový krok je 5 s – tzn. budu mít 21 řádků – 0 s až 100 s), počet sloupců je počet uzlů – délkových kroků (např. mám konstrukci tloušťky 300 mm a délkový krok bude 10 mm, pak budu mít 31 sloupců – 0 s až 300 mm). Uvnitř tabulky jsou teploty odpovídající času na řádku a umístění podle sloupečku. Teploty v aktuálním čase se vždy počítají z teplot v čase předchozím, a to ve stejném místě, v místě o 1 vpravo a o 1 vlevo. Z tohoto důvodu jsou potřeba 3 vzorečky. První vzoreček pro pravý sloupec, druhý vzoreček pro levý sloupec a třetí vzorec pro sloupečky uprostřed. Vzorečky pro krajní sloupce vycházejí ze vzorečku pro prostřední sloupce a jsou uvedeny níže.

Teploty v prvním řádku (značeny jako T_0) jsou počáteční teploty – je uvažován ustálený stav – běžně 20 °C.

		tloušťka konstrukce ->								
uzel		0	1	2	3	.	.	.	nx	
		$x = 0 \text{ m}$	$x = 1 * \Delta x \text{ m}$	$x = 2 * \Delta x \text{ m}$	$x = 3 * \Delta x \text{ m}$.	.	.	$x = nx * \Delta x = L \text{ m}$	
krok -< čas	0 $t = 0 \text{ s}$	T_0	T_0	T_0	T_0				T_0	T_0
	1 $t = \Delta t (5 \text{ s})$									
	2 $t = 2 * \Delta t (10 \text{ s})$									
	3 $t = 3 * \Delta t (15 \text{ s})$									
	.									
	.									
	nt $t = t_{lim} = nt * \Delta t (100 \text{ s})$									

Obr. 11 - Pomocný obrázek pro ukázkou výpočtu přestupu tepla v konstrukci

1.6.1 Rovnice pro výpočet teplot v prostředních buňkách

Z následující rovnice lze vypočítat teploty ve všech prostředních (modrých – dle Obr. 11) buňkách.

$$T_i^{k+1} = T_i^k + \frac{\lambda_i^k \cdot \Delta t}{\rho_i^k \cdot c_i^k} \cdot \left(\frac{T_{i+1}^k - T_{i-1}^k}{2 \cdot \Delta x} \right)^2 + \frac{\lambda_i^k \cdot \Delta t}{\rho_i^k \cdot c_i^k} \cdot \frac{T_{i+1}^k - 2 \cdot T_i^k + T_{i-1}^k}{(\Delta x)^2} \quad (67)$$

- T_i^{k+1} Teplota v aktuálním uzlu
- T_i^k Teplota ve stejném uzlu, v předchozím čase
- T_{i+1}^k Teplota ve vedlejším uzlu (pravém), v předchozím čase
- T_{i-1}^k Teplota ve vedlejším uzlu (levém), v předchozím čase

1.6.2 Rovnice pro výpočet teplot v krajních buňkách

Pro výpočet teplot v krajních buňkách (zelených a žlutých – dle Obr. 11) se výše uvedená rovnice musí upravit. Pro levý (zelený) sloupec se musí provést náhrada za T_{i-1}^k , protože vlevo od těchto buněk už žádné teploty v konstrukci nejsou. To samé platí pro pravý (žlutý) sloupec s tím, že se provede náhrada ze T_{i+1}^k .

Pro oba dva sloupečky budu potřebovat výpočet tepelného toku, ten budu počítat z normové okrajové podmínky, popsané výše (1.2.2.2). Vzoreček pro výpočet tepelného toku uvádím i zde:

$$q = \alpha_c \cdot (T_i^k - T_\infty) + \varepsilon \cdot \sigma \cdot (T_i^{k4} - T_\infty^4) \quad (68)$$

Náhrada pro levou stranu:

$$T_{i-1}^k = T_{i+1}^k - \frac{q_i^k \cdot 2 \cdot \Delta x}{\lambda} \quad (69)$$

Náhrada pro pravou stranu:

$$T_{i+1}^k = T_{i-1}^k - \frac{q_i^k \cdot 2 \cdot \Delta x}{\lambda} \quad (70)$$

Po dosazení do rovnice pro buňky uprostřed, a po všech úpravách, dostanu tyto rovnice:

- Pro levý sloupec ($i = 0$)

$$T_i^{k+1} = T_i^k + \frac{\lambda_i^k \cdot \Delta t}{\rho_i^k \cdot c_i^k} \cdot \left(\frac{q_i^k}{\lambda} \right)^2 + 2 \cdot \frac{\lambda_i^k \cdot \Delta t}{\rho_i^k \cdot c_i^k \cdot (\Delta x)^2} \cdot \left(-T_i^k + T_{i+1}^k - \frac{\Delta x \cdot q_i^k}{\lambda} \right) \quad (71)$$

- Pro pravý sloupec ($i = m$)

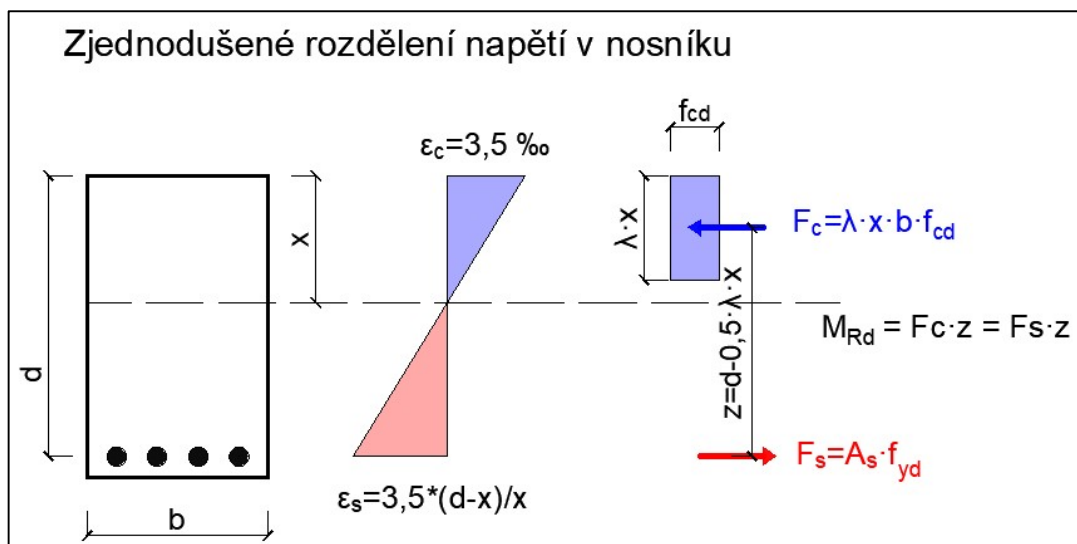
$$T_i^{k+1} = T_i^k + \frac{\lambda_i^k \cdot \Delta t}{\rho_i^k \cdot c_i^k} \cdot \left(-\frac{q_i^k}{\lambda} \right)^2 + 2 \cdot \frac{\lambda_i^k \cdot \Delta t}{\rho_i^k \cdot c_i^k \cdot (\Delta x)^2} \cdot \left(-T_i^k + T_{i-1}^k - \frac{\Delta x \cdot q_i^k}{\lambda} \right) \quad (72)$$

Tyto rovnice budu následně modelovat v programovacím jazyce python v praktické části (kapitola 2.3 Modelování 1D sdílení tepla pro betonové konstrukce).

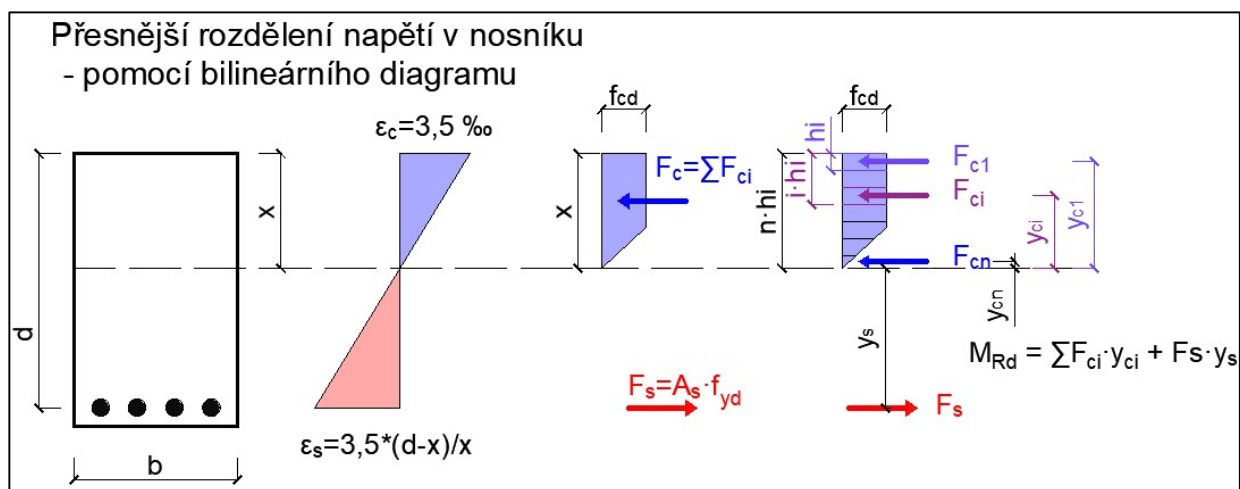
1.7 Proužková metoda

Proužková metoda se používá pro přesnější výpočet momentů únosnosti. Lze ji použít jak za běžné teploty, tak za požární situace. [8]

Za normálních okolností je počítán moment únosnosti se zjednodušeným rozložením napětí, které vypadá následovně (viz Obr. 12):



Je zde vidět zjednodušený (obdélníkový) tvar napětí s rozměry f_{cd} a $\lambda \cdot x$, většinou $0,8 \cdot x$. Tento tvar se odvíjí z pracovního diagramu pro beton a představuje přibližnou plochu pod křivkou z diagramu. Vzhledem k tomu, že se jedná o zjednodušený tvar, tudíž i zjednodušený výpočet, není konečný výsledek 100% přesný, ale je počítán s určitou chybou. K výpočtu přesných hodnot slouží tzv. proužková metoda, která rozdělí rozložení napětí na proužky a následně vypočítá síly a moment (viz Obr. 13).



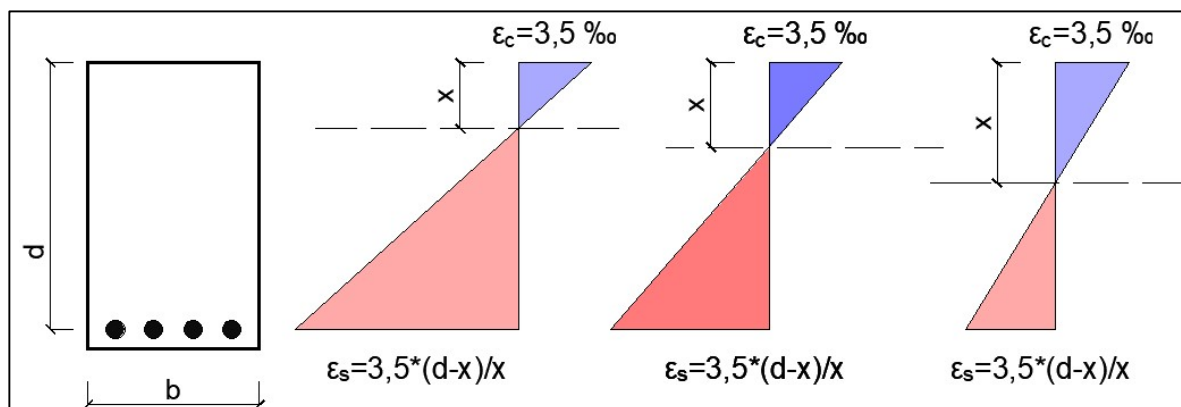
Na Obr. 13 je vidět, že rozložení napětí je podle tvaru pracovního bilineárního diagramu betonu. Toto napětí je rozděleno na n proužků o velikosti h_i . Platí, že čím víc je proužků a čím jsou proužky menší, tím přesnější bude výpočet. V každém jednotlivém proužku se provede výpočet síly F_{ci} , která se vypočítá vynásobením šířky proužku, příslušného napětí a šířkou prvku (nosníku). Pro každou sílu se určí vzdálenost od neutrální osy a následně se vypočítá moment, který je součtem násobků sil a jejich vzdáleností od neutrální osy. Současně se sleduje, aby výztuž byla za mezí kluzu. Pokud by výztuž nebyla za mezí kluzu a síly by se stále nerovnal, pak je výztuž v průřezu navržena špatně a musí se zkontrolovat konstrukční hodnoty návrhu výztuže (výztuž je předimenzována).

V případě výpočtu proužkovou metodou není známá poloha neutrální osy x . V tomto případě probíhají výpočty tak, že se jde po jednotlivých proužcích, vždy se spočítá celková síla v betonu a srovná se se silou v oceli. Tento postup se dělá tak dlouho, než si jsou síly rovny, v tuto chvíli lze zjistit polohu neutrální osy a následně lze dopočítat moment únosnosti průřezu (viz Obr. 14).

Příklad výpočtu:

- První proužek: $x = h_i (= 1 \text{ mm})$
 - o Výpočet: $F_{c1} = h_i \cdot b \cdot \sigma_{c1}$
 $F_c = F_{c1}$
 - o Porovnání: $F_s > F_c$
- Druhý proužek: $x = 2 h_i (= 2 \text{ mm})$
 - o Výpočet: $F_{c1} = h_i \cdot b \cdot \sigma_{c1}$
 $F_{c2} = h_i \cdot b \cdot \sigma_{c2}$
 $F_c = F_{c1} + F_{c2}$
 - o Porovnání: $F_s > F_c$
- N-tý proužek: $x = n h_i (= n \text{ mm})$
 - o Výpočet: $F_{c1} = h_i \cdot b \cdot \sigma_{c1}$
...
 $F_{cn} = h_i \cdot b \cdot \sigma_{cn}$
 $F_c = F_{c1} + \sum_{i=2}^n F_{ci}$
 - o Porovnání: $F_s = F_c$
 - o V tomto kroku víme polohu neutrální osy a provede se výpočet momentu
 - o Moment: $M_{Rd} = F_s \cdot (d - x) + \sum_{i=1}^n F_{ci} \cdot y_{ci}$

Takto vypočítaný moment by měl být přesnější než moment vypočítaný zjednodušenou metodou.



Obr. 14 - Proužková metoda, zjišťování osy x (Vytvořeno v AutoCADu)

V rámci praktické části provedu porovnání momentů vypočítaných proužkovou metodou a zjednodušenou metodou. Toto srovnání bude provedeno pomocí mnou vytvořeného kódu.

Proužkovou metodu lze stejným způsobem použít i za požární situace. Změna nastane ve funkci napětí, protože už to nebude funkce závislá pouze na pevnosti betonu, ale i na příslušné teplotě.

1.8 Výpočetní metody pro posouzení požární odolnosti ŽB desky

Pro posouzení požární odolnosti desky budu využívat zjednodušené výpočetní metody, tedy metodu izotermy 500°C a zónovou metodu. Údaje k těmto metodám jsou převzaty ze skript [9] a z normy ČSN EN 1992-1-2 [5].

1.8.1 Metoda izotermy 500 °C

Metoda izotermy 500 °C, zkráceně metoda IZO 500 je vhodná pro vystavení normového požáru a případně dalším časově závislým tepelným režimům, které způsobují podobná teplotní pole v prvku vystaveném požáru (např. pro parametrickou teplotní křivku, při uvažování součinitele otvorů $O \geq 0,14 m^{1/2}$).

Metoda je určena pro prvky zatížené ohybovým momentem a/nebo normálovou silou, jejichž průřezy splňují požadavky na minimální rozměry dle Tab. 1.

Tab. 1 - Minimální šířka průřezu

a) Vystavení normovému požáru

Požární odolnost	R 60	R 90	R 120	R 180	R 240
Min. šířka průřezu v mm	90	120	160	200	280

b) Vystavení parametrickému požáru

Hustota požárního zat. MJ/m ²	200	300	400	600	800
Minimální šířka průřezu v mm	100	140	160	200	240

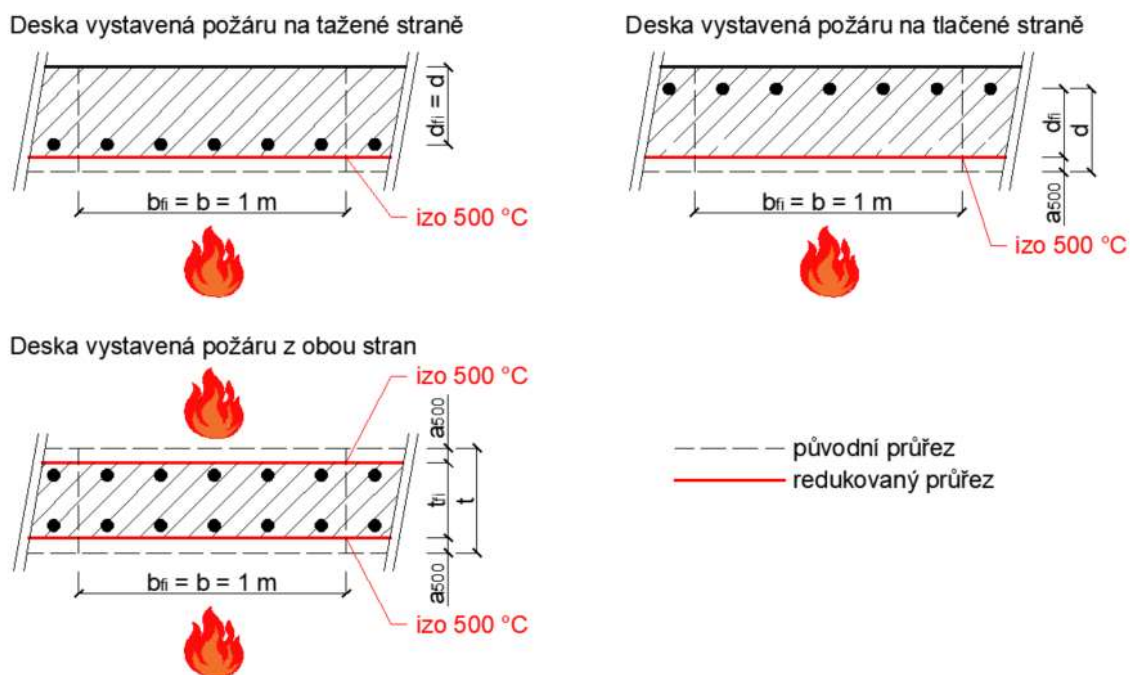
Předpoklady IZO500:

- Beton překračující teplotu 500 °C nepřispívá k únosnosti průřezu
- Beton do 500 °C má počáteční hodnoty pevnosti a modulu pružnosti
- Pevnost výztuže se redukuje úměrně její teplotě

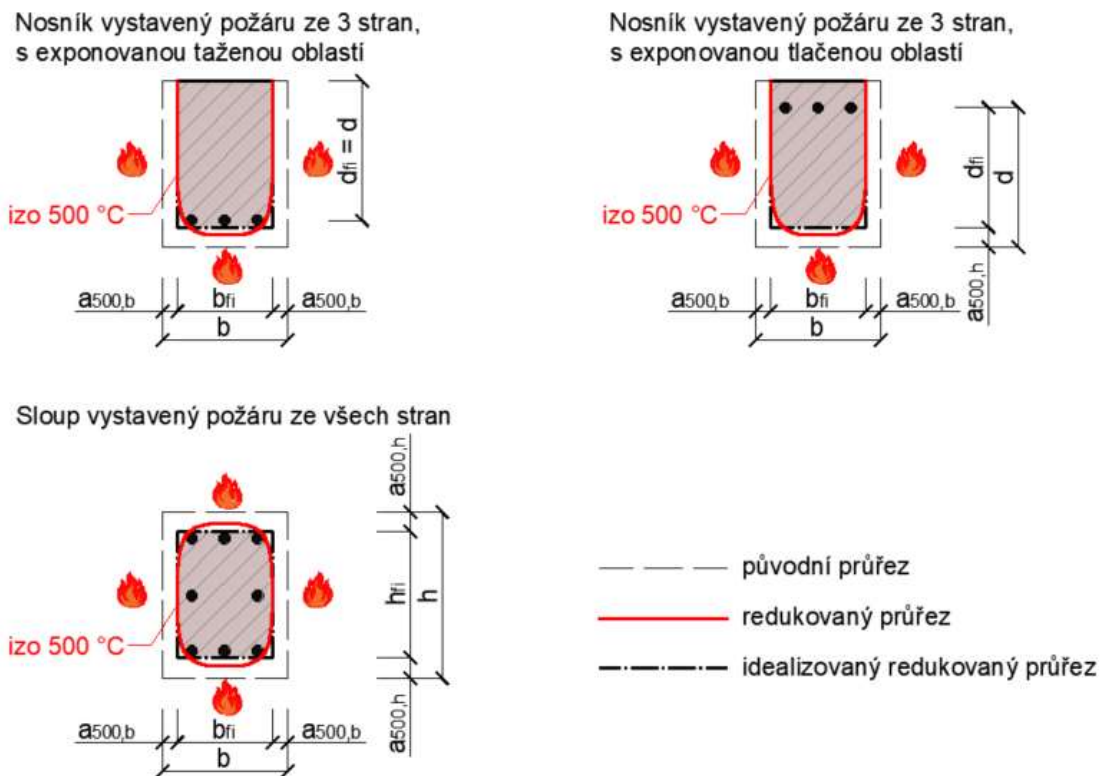
Postup posouzení:

- a) Stanovení rozložení teploty v průřezu
- b) Stanovení polohy izotermy 500 °C – definuje tvar a velikost redukovaného průřezu (viz Obr. 15 a Obr. 16)
 - Stanovení nové šířky b_{fi}
 - Stanovení nové výšky h_{fi}

- (Pro snazší výpočet lze uvažovat pravoúhlý tvar, místo zaobleného tvaru izotermy 500 °C. Plocha idealizovaného tvaru musí být shodná s plochou izotermy.)



Obr. 15 - Redukce průřezu desek a stěn při použití metody izotermy 500 °C (Vytvořeno v AutoCADu podle [9])



Obr. 16 - Redukce průřezu nosníků a sloupů při použití izotermy 500 °C (Vytvořeno v AutoCADu dle [9])

c) Výpočet pevnosti betonu v redukovaném průřezu

- Pevnost betonu v redukovaném průřezu je stejná jako při 20 °C
- Výpočet:

$$f_{cd,fi,20^{\circ}\text{C}} = \frac{f_{ck}}{\gamma_{c,fi}} \quad (73)$$

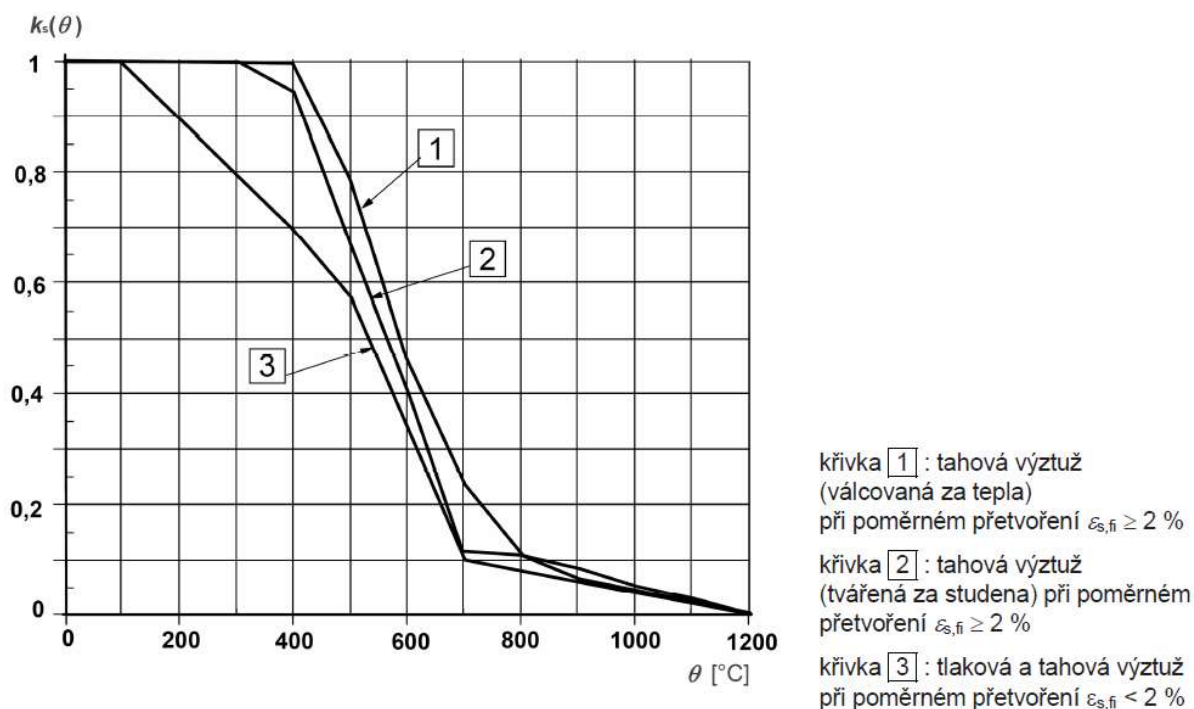
- f_{ck} = charakteristická hodnota pevnosti betonu při běžné teplotě
- $\gamma_{c,fi}$ = dílčí součinitel spolehlivosti betonu při požární situaci (=1,0)

d) Stanovení teploty v osách výztuže – lze uvažovat výztuž i mimo redukovaný průřez

e) Výpočet návrhové hodnoty pevnosti (resp. meze kluzu) výztuže při požáru

$$f_{syd,fi} = k_{s,\theta} \cdot \frac{f_{yk}}{\gamma_{s,fi}} \quad (74)$$

- f_{yk} = charakteristická hodnota meze kluzu betonářské výztuže při běžné teplotě
- $\gamma_{s,fi}$ = dílčí součinitel spolehlivosti výztuže při požární situaci (=1,0)
- $k_{s,\theta}$ = součinitel pro redukci charakteristické hodnoty meze kluzu betonářské výztuže odpovídající teplotě výztuže – viz Obr. 17 a Tab. 2

Obr. 17 - Součinitel $k_s(\theta)$ pro redukci charakteristické pevnosti (f_{yk}) tahové a tlakové výztuže (třída N)[5]

Tab. 2 – Součinitel $k_s(\Theta)$ pro redukci charakteristické pevnosti výztuže (Třídy N) v číslech – převzato z [5]

Teplota oceli Θ [°C]	$k_s(\Theta) = f_{sy,\Theta} / f_{yk} [-]$		
	1 – výztuž válcovaná za tepla	2 – výztuž tvářená za studena	3 – tlaková a tahová výztuž*
20	1,00	1,00	1,00
100	1,00	1,00	1,00
200	1,00	1,00	0,90
300	1,00	1,00	0,80
400	1,00	0,94	0,70
500	0,78	0,67	0,57
600	0,47	0,40	0,34
700	0,23	0,12	0,10
800	0,11	0,11	0,08
900	0,06	0,08	0,06
1000	0,04	0,05	0,04
1100	0,02	0,03	0,02
1200	0,00	0,00	0,00

*Bráno z rovnic níže – dle ČSN EN 1992-1-2 [5]

- Vztahy pro výztuž třídy N se smluvní mezí kluzu 0,2 nebo tahovou výztuž

$$\begin{aligned}
 k_{s,\theta} &= 1,0 && \text{pro } 20\text{ }^{\circ}\text{C} \leq \theta \leq 100\text{ }^{\circ}\text{C} \\
 k_{s,\theta} &= 0,7 - 0,3 \cdot \frac{\theta - 400}{300} && \text{pro } 100\text{ }^{\circ}\text{C} \leq \theta \leq 400\text{ }^{\circ}\text{C} \\
 k_{s,\theta} &= 0,57 - 0,13 \cdot \frac{\theta - 500}{100} && \text{pro } 400\text{ }^{\circ}\text{C} \leq \theta \leq 500\text{ }^{\circ}\text{C} \\
 k_{s,\theta} &= 0,1 - 0,47 \cdot \frac{\theta - 700}{200} && \text{pro } 500\text{ }^{\circ}\text{C} \leq \theta \leq 700\text{ }^{\circ}\text{C} \\
 k_{s,\theta} &= 0,1 \cdot \frac{1200 - \theta}{500} && \text{pro } 700\text{ }^{\circ}\text{C} \leq \theta \leq 1200\text{ }^{\circ}\text{C}
 \end{aligned} \tag{75}$$

- Pokud mají výztužné pruty různé teploty, stanoví se průměrná pevnost jednotlivých prutů dle bodu f)

f) Stanovení průměrné pevnosti prutů

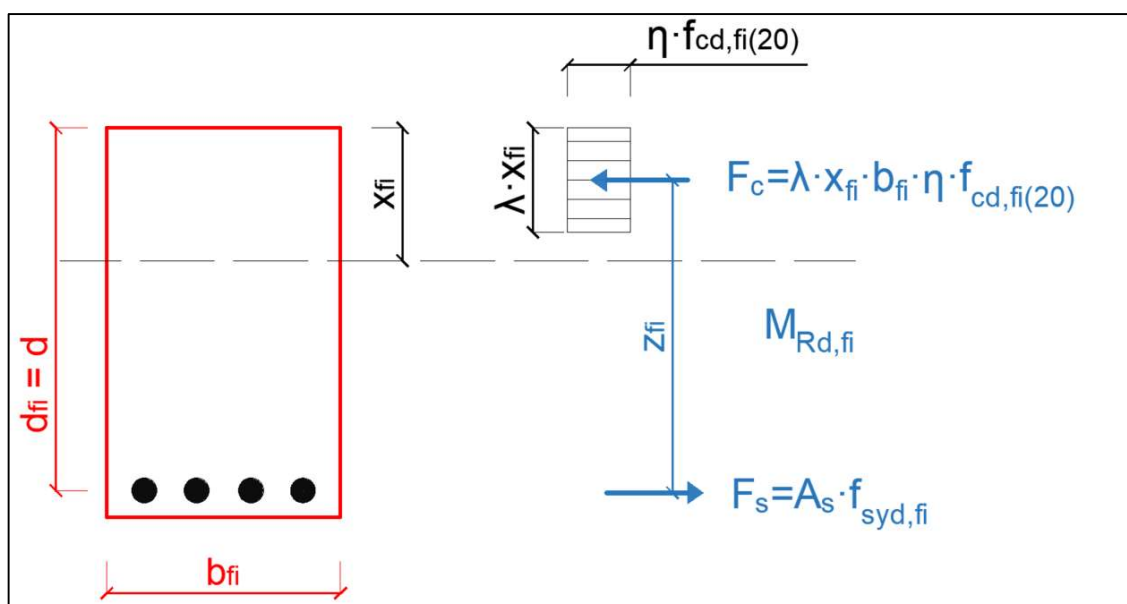
$$f_{syd,fi,v} = k_{s,v} \cdot \frac{f_{yk}}{\gamma_{s,fi}} \tag{76}$$

- $k_{s,v}$ = průměrný redukční součinitel pro v-tou vrstvu výztuže

$$k_{s,v} = \frac{\sum_{i=1}^n k_{s,\theta_i}}{n} \tag{77}$$

- n je celkový počet prutů ve „v“ vrstvě

g) Stanovení návrhové hodnoty únosnosti s uvažáním příslušných materiálových charakteristik (Obr. 18)



Obr. 18 - Rozdělení napětí v obdélníkovém průřezu při požární situaci (Vytvořeno v AutoCADu dle [9])

$$x_{fi} = \frac{A_s \cdot f_{syd,fi,v}}{b_{fi} \cdot \lambda \cdot \eta \cdot f_{cd,fi,20^\circ C}} \quad (78)$$

- Návrhová hodnota momentu únosnosti při požární situaci:

$$M_{Rd,fi} = A_s \cdot f_{syd,fi,v} \cdot z_{fi} = A_s \cdot f_{syd,fi,v} \cdot (d_{fi} - 0,5 \cdot \lambda \cdot x_{fi}) \quad (79)$$

h) Porovnání návrhové hodnoty mezní únosnosti průřezu s odpovídajícím účinkem hodnoty zatížení při požární situaci

- Porovnání z hlediska únosnosti:

$$R_{d,fi,t} \geq E_{d,fi,t} \quad (80)$$

- $R_{d,fi,t}$ = Návrhová hodnota únosnosti prvku při požární situaci ($=M_{Rd,fi}$)
- $E_{d,fi,t}$ = Návrhová hodnota příslušných účinků zatížení při požáru, v čase t

$$E_{d,fi,t} = E_{d,fi} = \eta_{fi} \cdot E_d \quad (81)$$

- E_d = návrhová hodnota účinků zatížení za běžné teploty
- η_{fi} = redukční součinitel (konzervativně lze uvažovat $\eta_{fi} = 0,7$ pro betonové konstrukce)

$$\eta_{fi} = \frac{G_k + \psi_{fi} \cdot Q_{k,1}}{\gamma_G \cdot G_k + \gamma_{Q,1} \cdot \psi_{0,1} \cdot Q_{k,1}} \quad (82)$$

1.8.2 Zónová metoda

Zónová metoda je dle ČSN EN 1992-1-2 [5] vhodná pro vystavení normového požáru a případně dalším časově závislým tepelným režimům, které způsobují podobná teplotní pole v prvku vystaveném požáru.

Metoda je určena pro prvky zatížené ohybovým momentem a/nebo normálovou silou a poskytuje přesnější výsledky než izoterma 500 °C, zvláště pro sloupy.

Odlišnost od metody IZO500:

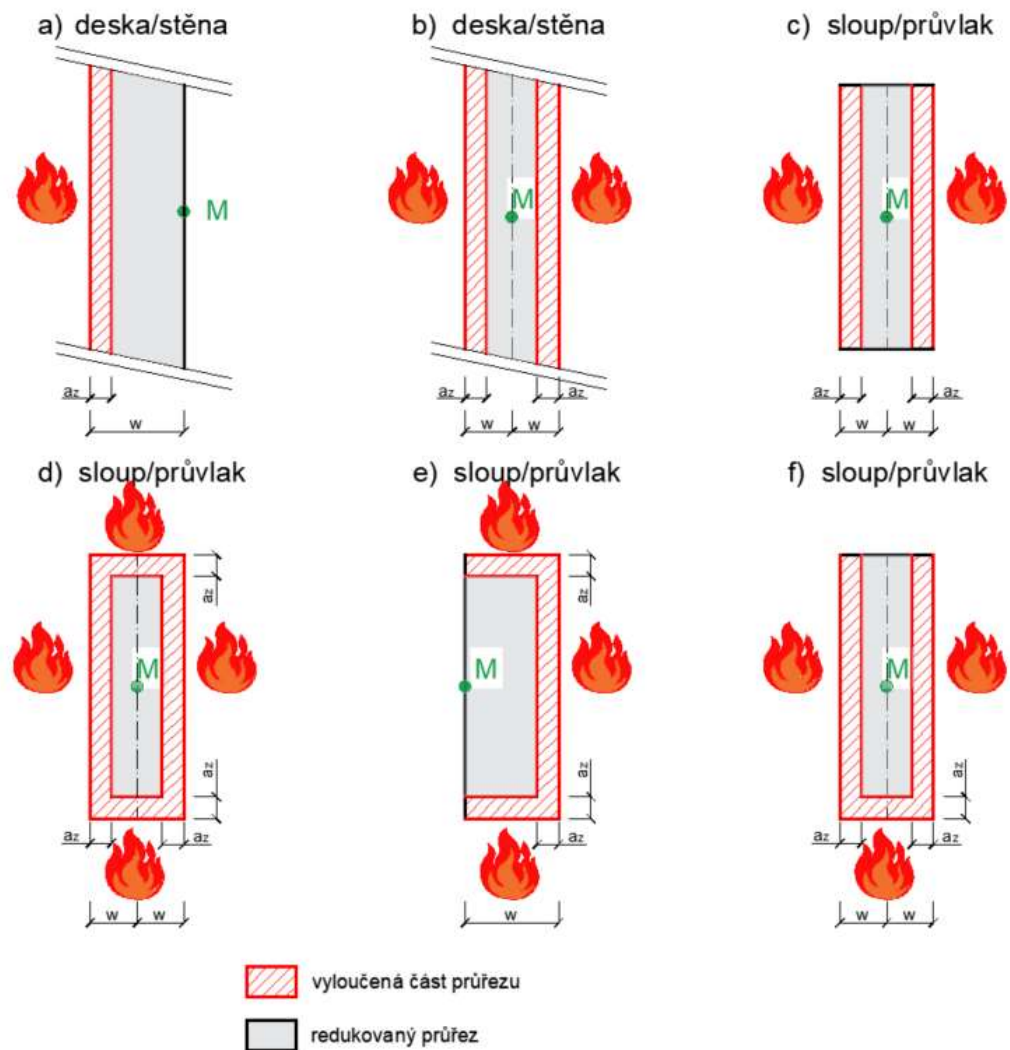
- Jiné stanovení tloušťky vyloučené vrstvy betonu
- Redukce pevnosti betonu

Předpoklady Zónové metody:

- Pevnost betonu se redukuje úměrně k jeho teplotě
- Pevnost výztuže se redukuje úměrně její teplotě

Postup posouzení:

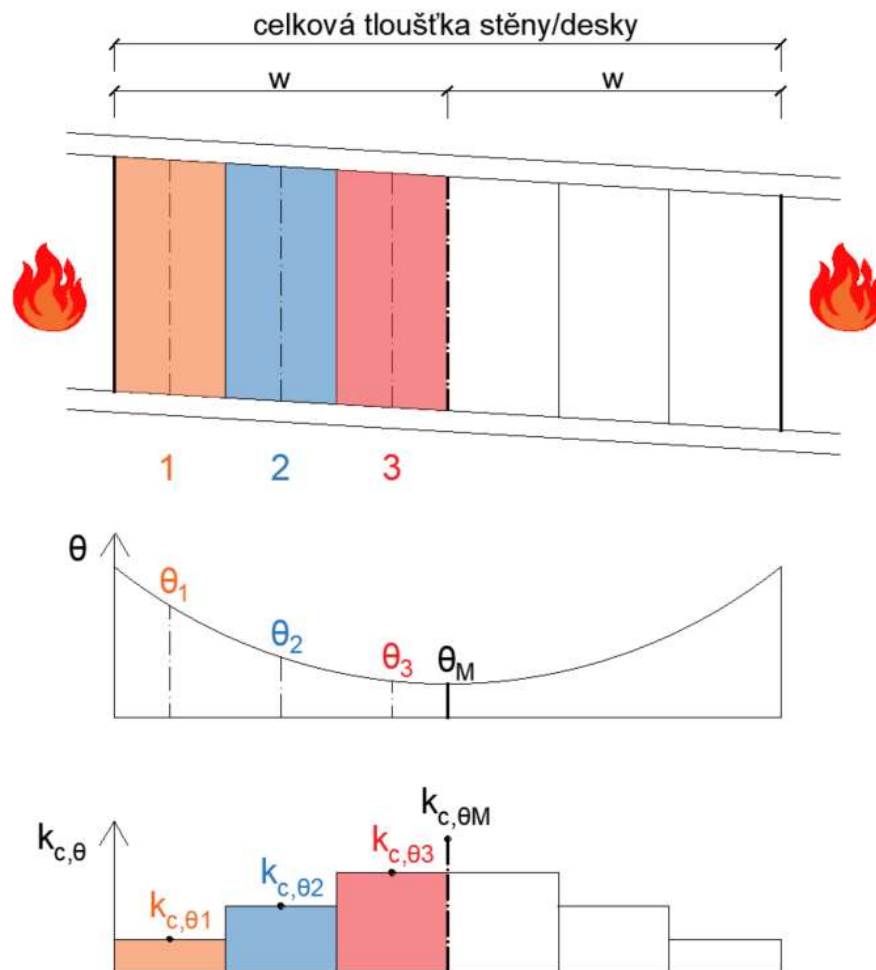
- a) Stanovení charakteristické tloušťky „ w “ a polohy bodu M
 - Deska nebo stěna vystavená požáru z 1 strany (viz Obr. 19a):
 - w = tloušťka průřezu
 - Deska nebo stěna vystavená požáru ze 2 stran (viz Obr. 19b):
 - w = 1/2 tloušťky průřezu
 - Obdélníkový průřez vystavený požáru ze 2 protilehlých stran (viz Obr. 19c):
 - w = 1/2 vzdálenosti mezi těmito stranami
 - Obdélníkový průřez vystavený požáru ze 4 stran (viz Obr. 19d):
 - w = 1/2 menšího z rozměrů průřezu
 - Na všech stranách průřezu se uvažuje tl. a_z stanovená pro tl. w
 - Obdélníkový průřez vystavený požáru ze 3 stran (viz Obr. 19e,f):
 - w = menší z:
 - Vzdálenost mezi stranou nevystavenou a protilehlou vystavenou požáru
 - 1/2 vzdálenosti mezi protilehlými stranami průřezu vystavených požáru
 - Na všech stranách průřezu vystavených požáru se uvažuje tl. a_z stanovená pro tl. w



Obr. 19 - Redukce průřezu při použití zónové metody ((Vytvořeno v AutoCADu dle [9])

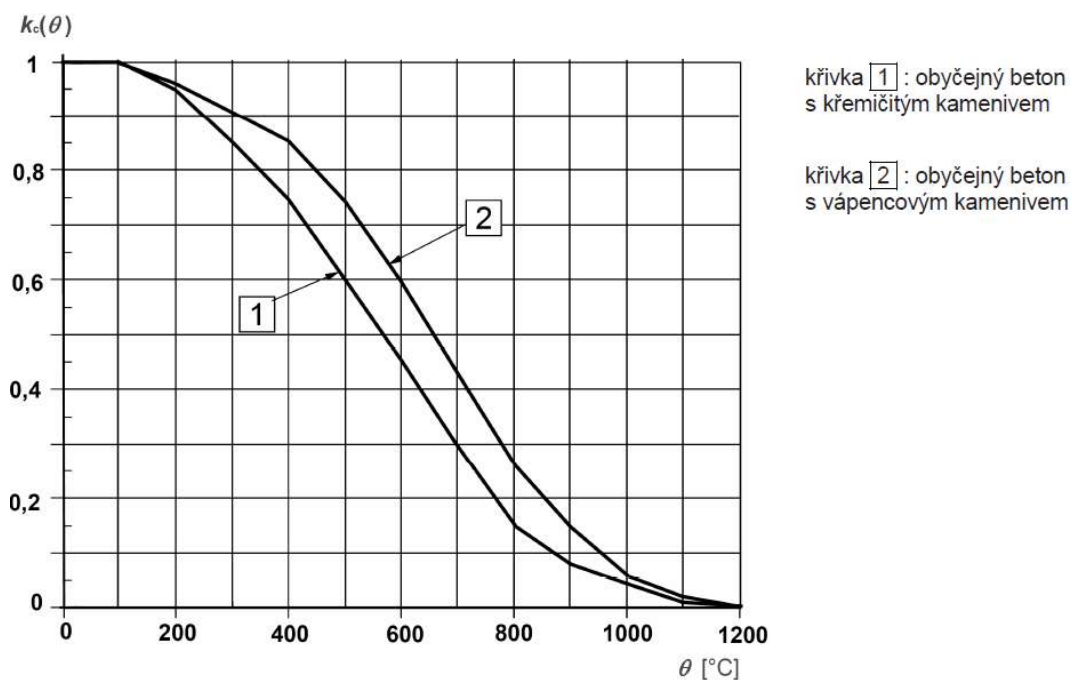
b) Rozdělení tloušťky „ w “ na „ n “ rovnoběžných zón o stejné tloušťce

- Zóny musí být vždy alespoň 3 ($n \geq 3$)
- Pro každou zónu se následně určí teplota a redukční součinitel – viz následující body a Obr. 20



Obr. 20 - Rozdělení průřezu na zóny, stanovení teplot a redukčních součinitelů (Vytvořeno v AutoCADu dle [9])

- c) Stanovení teploty θ_i ve středu každé zóny a teploty θ_M v bodě M
- d) Stanovení součinitele $k_{c,\theta i}$ pro redukci charakteristické hodnoty pevnosti betonu v tlaku, odpovídající teplotě θ_i a $k_{c,\theta M}$ pro redukci v bodě M dle teploty θ_M
 - Součinitel se stanoví dle Obr. 21 nebo dle Tab. 3



Obr. 21 - Součinitel $k_{c\theta}$ pro redukci charakteristické hodnoty pevnosti betonu v tlaku při zvýšených teplotách [5]

Tab. 3 - Součinitel pro redukci charakteristické hodnoty pevnosti v tlaku při zvýšených teplotách v číslech (převzato z [5])

Teplota [$^{\circ}\text{C}$]	Beton s křemičitým kamenivem	Beton s vápencovým kamenivem
20	1,00	1,00
100	1,00	1,00
200	0,95	0,97
300	0,85	0,91
400	0,75	0,85
500	0,60	0,74
600	0,45	0,60
700	0,30	0,43
800	0,15	0,27
900	0,08	0,15
1000	0,04	0,06
1100	0,01	0,02
1200	0,00	0,00

e) Stanovení středního součinitele redukce $k_{c,m}$

$$k_{c,m} = \frac{1 - \frac{0,2}{n}}{n} \cdot \sum_{i=1}^n k_{c,\theta i} \quad (83)$$

- n = počet zón
- i = číslo zóny

f) Stanovení tloušťky vyloučené vrstvy a_z , která definuje rozměry redukovaného průřezu

- Pro nosníky, desky a smykové stěny **bez účinků** 2.řádu platí:

$$a_z = w \cdot \left(1 - \frac{k_{c,m}}{k_{c,\theta M}}\right) \quad (84)$$

- Pro sloupy, stěny a jiné konstrukce **s účinky** 2.řádu platí:

$$a_z = w \cdot \left[1 - \left(\frac{k_{c,m}}{k_{c,\theta M}}\right)^{1/3}\right] \quad (85)$$

g) Redukce pevnosti betonu v závislosti na teplotě θ_M

$$f_{cd,fi} = k_{c,\theta M} \cdot \frac{f_{ck}}{\gamma_{c,fi}} \quad (86)$$

- f_{ck} = charakteristická pevnost betonu v tlaku při běžné teplotě
- $\gamma_{c,fi}$ = dílčí součinitel spolehlivosti betonu při požáru ($\gamma_{c,fi} = 1$)

h) Stanovení teploty výztuže a její redukce (stejně jako v metodě IZO 500 °C)

- Viz bod e) v metodě izotermy 500 °C

i) Stanovení návrhové hodnoty únosnosti redukovaného průřezu

- Viz bod g) v metodě izotermy 500 °C

j) Porovnání návrhové hodnoty mezní únosnosti průřezu s účinkem návrhové hodnoty zatížení při požární situaci

- Viz bod h) v metodě izotermy 500 °C

1.9 Dostupné programy

Zde uvedu programy, které řeší prostup tepla konstrukcí a napíši k nim možné výhody a nevýhody na základě subjektivního názoru.

Programy TeDisOn [1P], RCC_{fi} [2P] a FiDeS [3P] vznikly na ČVUT, Fakultě stavební na katedře betonových a zděných konstrukcí a na všech těchto programech se podílel Ing. Radek Štefan, Ph.D. Tyto programy jsou volně dostupné na internetu.

Dalším programem, který zde uvedu, je software ANSYS [4P], který byl vyvinut firmou Ansys, Inc. a je dostupný pouze po zakoupení licence.

Existuje i mnoho programů na prostup tepla, jedná se ale o programy, které řeší problém za běžné teploty – tzn. zkoumají prvky hlavně pro posouzení tepelné obálky budovy. Tyto programy zde nebudu uvádět, protože zde se zabývám vedením tepla za požáru, a to je dost odlišné od vedení tepla za běžné teploty (např. co se týče změny materiálových vlastností za teploty, zadání požárních křivek atd.)

1.9.1 TeDisOn

Jedná se o program zvaný Temperature Distribution Online, který vytvořil Radek Štefan a Jakub Holan v roce 2019 [10]. Tento program je pouze v anglické verzi.

Tento program se věnuje výpočtu 1D teplotního rozložení teploty v betonové zdi nebo desky během požáru. Výstupem z tohoto programu je graf znázorňující rozložení teploty po tloušťce po uplynutí stanoveného období, kdy je prvek vystaven ohni.

Zadává se zde pouze tloušťka konstrukce, trvání ohně, expozice, počáteční hustota, počáteční objem vody a tepelná vodivost (Obr. 22).

Výhody programu:

- je on-line, tzn. není zde potřeba žádná instalace
- je jednoduchý na ovládání
- vhodný pro rychlé ověření teploty
- pěkný interaktivní graf, který ukazuje teplotu ve kterékoli části konstrukce
- rychlá odezva programu

Nevýhody programu:

- teplota je zadána pouze nominální normovou křivkou
- funguje jen pro betonové desky nebo zdi
- není dostupný v české verzi

TeDisOn - Temperature Distribution Online

A program for a 1D calculation of temperature distribution over a concrete wall or slab during fire using the FDM

Author: Radek Štefan, Jakub Holan

Input parameters

Thickness: <input type="text" value="0,2"/> m	Initial density: <input type="text" value="2400"/> kg/m ³
Duration of fire: <input type="text" value="30"/> minutes	Initial water content: <input type="text" value="1,5"/> %
Exposure ^[1] : <input type="text" value="two-sided"/> ▾	Thermal conductivity: <input type="text" value="upper limit"/> ▾

Obr. 22 - Vstupní data programu TeDisOn [1P]

1.9.2 RCC_{fi}

RCC_{fi} je výpočetní program pro posouzení požární odolnosti železobetonových sloupů metodou popsanou v příloze B.3 normy ČSN EN 1992-1-2. Tento program vytvořil Josef Sura, Radek Štefan a Jaroslav Procházka [2P].

Je to program, který je volně ke stažení, ale k jeho používání musí být instalována i knihovna MATLAB Compiler Runtime 9.0. Tento program je dostupný v jazyce anglickém i českém a slouží pro výpočet sloupů obdélníkového průřezu nebo průřezu tvaru I. Výstupem z tohoto programu je rozložení teploty po průřezu a také protokoly, ve kterých je uvedena teplota výztuže a spočítán moment II. Řádu, který slouží pro porovnání požární odolnosti.

Do RCC_{fi} vstupují údaje o rozměrech sloupu, včetně umístění výztuže, dále zatížení na sloup včetně excentricity, doba vystavení požáru a údaje o materiálu (Obr. 23).

Výhody programu:

- graf rozložení teploty v průřezu
- intuitivní ovládání + vysvětlující popis k jednotlivým veličinám
- dána přesná teplota ve výztuži
- kromě teplotního profilu vypočítá i moment II. řádu a posoudí požární odolnost

Nevýhody programu:

- teplota je zadána pouze nominální normovou křivkou
- funguje jen pro betonové sloupy s obdélníkovým nebo I průřezem
- je k němu potřeba knihovna Matlab
- má dlouhý výpočetní čas

RCCfi 1.2 - Preprocessor

Rozměry

b [mm] 300 ?

h [mm] 300 ?

$l_{0,fi}$ [mm] 3000 ?

ϕ [mm] 14 ?

a [mm] 40 ?

$a_1, a_2, a_3, \dots, a_n$ [mm]

40,95,150 ?

Zatížení

$N_{Ed,fi}$ [kN] 500 ?

$e_{0,fi}$ [mm] 40 ?

c [-] 10 ?

Vystavení požáru (ISO křivka)

t [min] 30 ?

Materiály

Třída betonu C25/30 ?

ρ_{20} [kg m^{-3}] 2300 ?

u [%] 1.5 ?

λ Dolní mez ?

f_{yk} [MPa] 500 ?

Schéma průřezu

$N_{Ed,fi}$

$e_{0,fi}$

C_g

d

h

a_1

a_2

a_3

b

VÝPOČET

VYKRESLIT PRŮŘEZ **NOVÝ** **UKONČIT**

Obr. 23 - Vstupní údaje z programu RCCfi [2P]

1.9.3 FiDeS

FiDeS (= Fire Design Software) je soubor výpočetních programů pro navrhování betonových a zděných konstrukcí na účinky požáru podle Eurokódů vytvořený Ing. Radkem Štefanem v letech 2010-2016 [3P].

Je to soubor programů, který je volně ke stažení, ale k jeho používání musí být instalována i knihovna MATLAB Compiler Runtime 9.0. Tento program je dostupný v českém jazyce (pouze titulní stránka) a následně v jazyce anglickém.

Soubor zahrnuje celkem 5 samostatných programů:

- program pro teplotní analýzu požárního úseku pomocí nominálních teplotních křivek a pomocí parametrické teplotní křivky
- Temp Analysis - program pro teplotní analýzu obdélníkových průřezů z různých stavebních materiálů vystavených normovému nebo parametrickému požáru
- program pro ověření normové požární odolnosti vybraných betonových prvků
- program pro stanovení normové požární odolnosti vybraných betonových prvků (sloupů) pomocí zjednodušených výpočetních metod
- program pro posouzení normové požární odolnosti vybraných zděných prvků (stěn)

1.9.3.1 TempAnalysis

Pro tuto práci mě bude zajímat pouze 2. program a to program TempAnalysis pro teplotní analýzu průřezu, který řeší 1D i 2D problémy. Tento program vytvořil Radek Štefan a Jaroslav Procházka v letech 2009-2016.

Do tohoto programu vstupují údaje o materiálu, tloušťce, ochranné vrstvě, požární křivce a době expozice (Obr. 24).

Výstupem je teplotní profil a výpočet teploty v daném místě.

Výhody programu:

- schéma rozložení teploty v průřezu na základě podrobných informací
- intuitivní ovládání
- lze dopočítat teplotu v jakémkoli místě
- lze uvažovat izolační vrstvu
- srozumitelná a velmi podrobná příručka

Nevýhody programu:

- teplota je zadána buď nominální normovou křivkou nebo parametrickou, ale pouze se zadáním 4 parametrů ($q_{t,d}$, O , b a rychlost rozvoje požáru)
- funguje jen pro obdélníkové průřezy
- je k němu potřeba knihovna Matlab
- má delší výpočetní čas
- program je pouze v anglickém jazyce



Obr. 24 - Vstupní údaje programu TempAnalysis [3P]

1.9.4 ANSYS

Software ANSYS [4P] je také vhodným programem k modelování prostupu tepla konstrukcí. ANSYS programů je hned několik, problém je v tom, že na využívání těchto programů je potřeba zakoupení licence.

Do programu se dá zadat jakýkoli tvar konstrukce a jakýkoli požár. Podmínkou je, umět „programovat“ a umět anglicky. Software funguje na principu zadávání příkazů do příkazového řádku, popř. přepínání ve „stromu funkcí“.

Výsledkem je barevné schéma rozložení teplot v konstrukci.

Výhody programu:

- Lze zadat jakýkoli tvar konstrukce a jakákoli teplotní křivka
- Dostupná „knihovna“ příkazů na internetu

Nevýhody programu:

- Nutné zakoupení licence
- Složitě ovládání (pouze přes příkazový řádek, neumí funkci „zpět“, velké množství příkazů, každý příkaz se musí dohledávat na internetu)
- Složitě zadávání teplot (je k tomu potřeba vytvořit textový soubor a pomocí příkazů psát jednotlivé teploty v konkrétním čase, popř. lze zadat rovnice křivky)
- Program dostupný pouze v angličtině

1.10 Programovací jazyk

Pro tuto práci se budu pohybovat v prostředí programovacích jazyků Matlab-Python [5P]. Zde vám uvedu nějaké informace o tom, co to vlastně Matlab a Python je a také jak pracují a jaký je v nich rozdíl.

1.10.1 Matlab

Matlab je komerční numerické počítačové prostředí a programovací jazyk. Standardní základní knihovna neobsahuje tolik funkcí jako Python, ale zahrnuje maticovou algebru a rozsáhlou knihovnu pro zpracování a vykreslování dat [5P].

Výhody: množství funkcí; balíček obsahuje vše, co je potřeba a nemusí se instalovat další balíčky; velká vědecká komunita.

Nevýhody: vysoká cena; algoritmy jsou proprietární (nejsou vidět kódy algoritmů); omezuje přenositelnost kódu (instalovaná verze se musí shodovat s verzí, ve které to bylo vytvořeno).

1.10.2 Python

Python je programovací jazyk, který také obsahuje rozsáhlé standardní knihovny (pro vytváření sítí, databází, ..). K provádění výpočtů bude potřeba dalších balíčků, např. Numpy (pro práci s maticemi a vícerozměrnými poli), Matplotlib (pro vykreslování výsledků) a další balíčky dle potřeby uživatele. Rozdíl oproti Matlabu je také v tom, že Matlab používá indexování od 1, ale python od 0 [5P].

Výhody: program je zdarma; snadno čitelný jazyk; rychlost spouštění; snadná manipulace s řetězci; přenositelnost (kód se dá spouštět všude); lze definovat libovolné množství funkcí.

Kapitola 2

Praktická část

Praktická část je rozdělena na 3 bloky, přičemž každý je věnován popisu jiné problematiky.

1. První blok se věnuje sestavení kódu pro modelování 1D přestupu tepla skrz konstrukce.
2. Druhý blok se věnuje sestavení kódu pro výpočet momentu únosnosti pomocí proužkové metody.
3. Třetí blok se věnuje sestavení kódu pro posouzení požární odolnosti podle metody izotermy 500 °C a zónové metody.

2 Praktická část – 1. blok

V 1. bloku praktické části se budu věnovat modelování 1D přestupu tepla skrz konstrukce. Popíšu zde podrobné sestavení kódu, od zadávání jednotlivých křivek, přes materiálové vlastnosti až k samotnému kódu výpočtu rozložení teploty.

Uvedu zde kódy z programovacího jazyka pro teplotní křivky, materiálové charakteristiky a následně pro 1D přestup tepla skrz konstrukci. Veškeré modelování jsou vytvořeny podle rovnic popsaných v teoretické části této diplomové práce.

V závěru provedu porovnání výsledků z mého kódu s výsledky některých programů, popsaných výše.

2.1 Teplotní křivky

Teplotní křivky jsou základním vstupem pro kód přestupu tepla, proto zde uvedu svůj kód z programovacího jazyka Python (Obr. 25), který popisuje rovnice pro Nominální teplotní křivky.

```
# zadání zkoumaného času intervalu od, do(včetně) a po jakém kroku v sekundách
od= 0
do= 6000
krok= 5

# přepočítané indexy na minuty od1, do1, krok1
od1= od/60
do1= (do+1)/60
krok1= krok/60

time = np.arange(od1,do1,krok1)
# print(time)

def Normova_krivka(t):
    T = 20+345*np.log10(8*t+1)
    return T

def Uhlovodikova_krivka (t):
    T = 20+1080*(1-0.325*(e**(-0.167*t))-0.675*(e**(-2.5*t)))
    return T

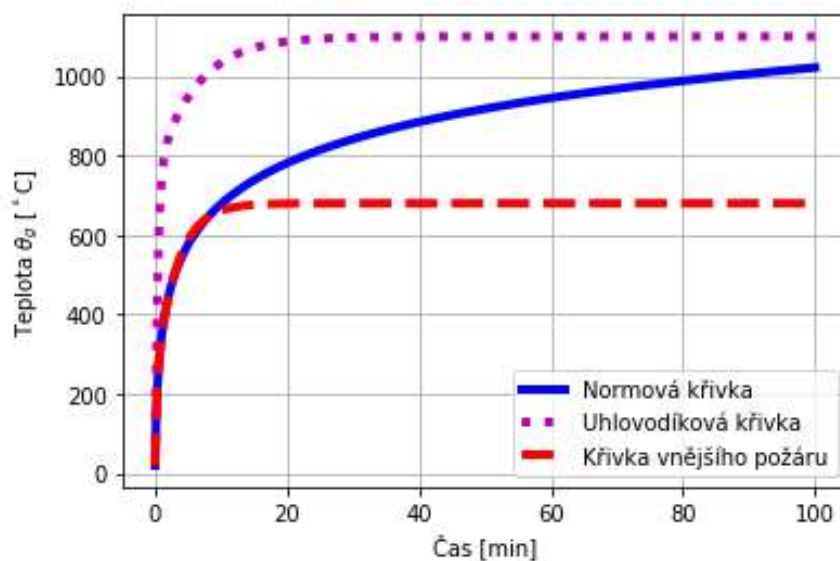
def Krivka_vnejsiho_pozaru (t):
    T = 20+660*(1-0.687*(e**(-0.32*t))-0.313*(e**(-3.8*t)))
    return T

normova= []
uhlovodikova= []
vnejsiho_pozaru = []

for t in time:
    normova.extend([Normova_krivka(t)])
    uhlovodikova.extend([Uhlovodikova_krivka(t)])
    vnejsiho_pozaru.extend([Krivka_vnejsiho_pozaru(t)])
```

Obr. 25 - Nominální teplotní křivky (Vytvořeno v prostředí Python podle [1])

Výstupem z kódu uvedeného výše jsou jednotlivé křivky - viz Obr. 26:



Obr. 26 - Nominální teplotní křivky -graf (Vytvořeno v prostředí Python podle [1])

2.2 Materiálové vlastnosti betonu

Zde uvedu vstupy a výstupy pro materiálové vlastnosti betonu, které jsou proměnné s teplotou. Rovnice viz teoretická část této práce.

2.2.1 Měrné teplo

Vstup: Zadání jednotlivých křivek – pro beton suchý a beton s 1,5 % a 3 % vlhkostí (Obr. 27 a Obr. 28).

```
def Merne_teplo_0 (T): # J / kg*K
    if T >=20 and T<100:
        cp= 900
    elif T >=100 and T<200:
        cp= 900 + T-100
    elif T >=200 and T<400:
        cp= 1000 + (T-200)/2
    elif T >=400 and T<=1200:
        cp= 1100
    else:
        cp= 1100
    return (cp)
```

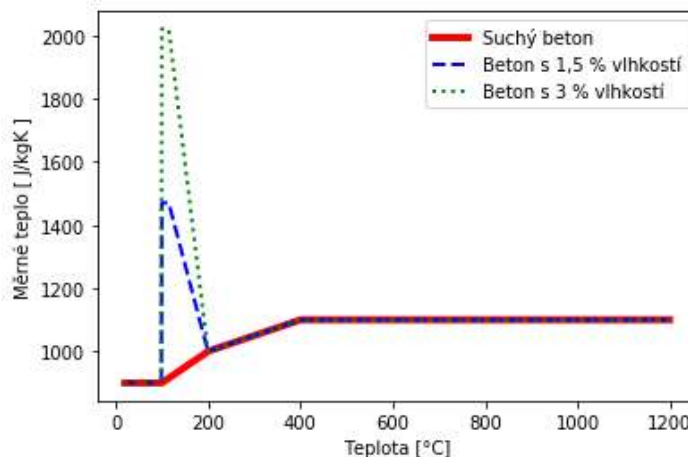
Obr. 27 - Měrné teplo - Beton suchý (Vytvořeno v prostředí Python podle [5])

```
def Merne_teplo_1 (T): # J / kg*K
    if T >=20 and T<100:
        cp1= 900
    elif T >=100 and T<115:
        cp1= 1470
    elif T >=115 and T<200:
        cp1= 1470 - (T-115)*94/17
    elif T >=200 and T<400:
        cp1= 1000 + (T-200)/2
    elif T >=400 and T<=1200:
        cp1= 1100
    else:
        cp1= 1100
    return (cp1)
```

```
def Merne_teplo_3 (T): # J / kg*K
    if T >=20 and T<100:
        cp3= 900
    elif T >=100 and T<115:
        cp3= 2020
    elif T >=115 and T<200:
        cp3= 2020 - (T-115)*12
    elif T >=200 and T<400:
        cp3= 1000 + (T-200)/2
    elif T >=400 and T<=1200:
        cp3= 1100
    else:
        cp3= 1100
    return (cp3)
```

Obr. 28 - Měrné teplo - Beton s 1,5 a 3 % vlhkostí (Vytvořeno v prostředí Python podle [5])

Výstupem je graf křivek měrného tepla, který zde uvádím pro možnost srovnání s obrázky uvedené v příslušných normách (Obr. 29).



Obr. 29 - Měrné teplo - graf (Vytvořeno v prostředí Python podle [5])

2.2.2 Objemová hmotnost

Vstup (Obr. 30):

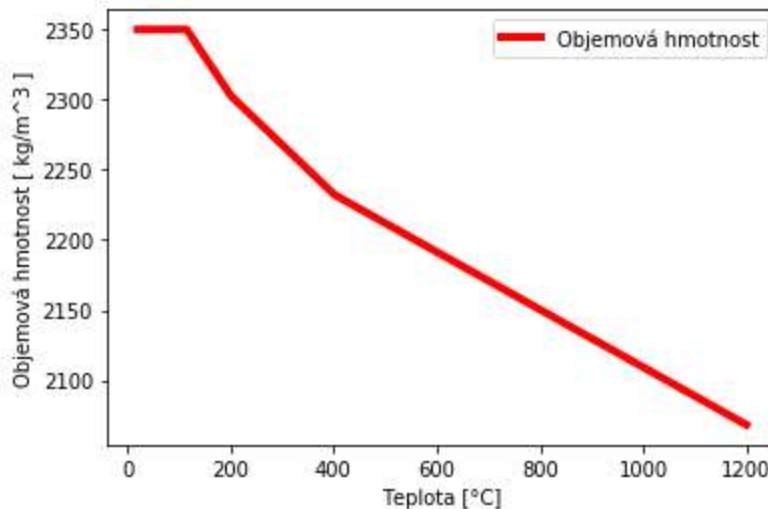
```
'''Objemová hmotnost'''
def Objemova_hmotnost (T,ro20):          # kg / m3
    if T >=20 and T<115:
        ro = ro20
    elif T >=115 and T<200:
        ro = ro20*(1-0.02*(T-115)/85)
    elif T >=200 and T<400:
        ro = ro20*(0.98-0.03*(T-200)/200)
    elif T >=400 and T<=1200:
        ro = ro20*(0.95-0.07*(T-400)/800)
    else:
        ro= ro20*(0.95-0.07*(1200-400)/800)
    return (ro)

objemova_hmotnost=[]
for teplota in T:
    objemova_hmotnost.append([Objemova_hmotnost (teplota,ro20)])
"Vykreslení"
print (objemova_hmotnost)
plt.plot(T,objemova_hmotnost,'r',linewidth=4,label='Objemová hmotnost')

plt.xlabel('Teplota [°C]')
plt.ylabel('Objemová hmotnost [ kg/m3 ]')
plt.legend()
```

Obr. 30 – Objemová hmotnost (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 31):



Obr. 31 – Objemová hmotnost - graf (Vytvořeno v prostředí Python podle [5])

2.2.3 Součinitel tepelné vodivosti

Vstupy (Obr. 32):

```
'''Součinitel tepelné vodivosti- Beton, Horní mez'''
def Tepelna_vodivost0 (T):          # W / m*K
    lam= 2-0.2451*(T/100)+0.0107*((T/100)**2)
    return (lam)

Vodivost_h=[]
for teplota in T:
    Vodivost_h.append([Tepelna_vodivost0(teplota)])
"Vykreslení"
print (Vodivost_h)
plt.plot(T,Vodivost_h,'r',linewidth=4,label='Horní mez')

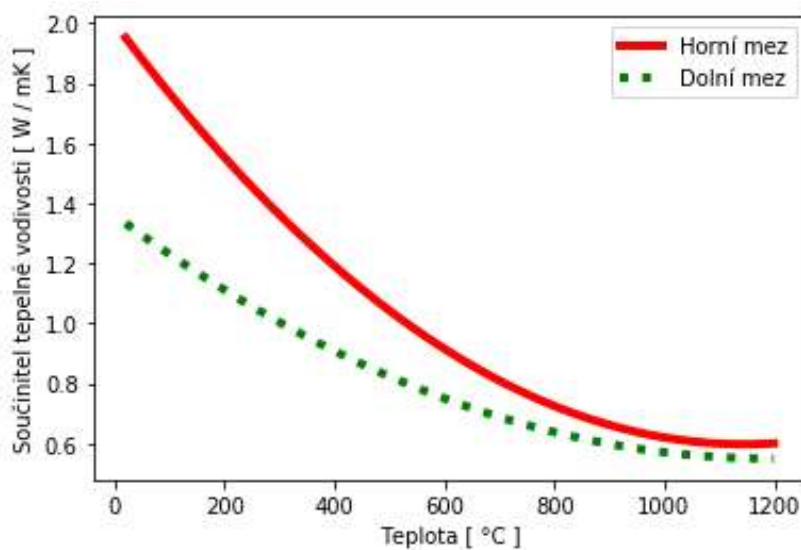
# # #####
# # #####
'''Součinitel tepelné vodivosti- Beton, Doní mez'''
def Tepelna_vodivost1 (T):          # W / m*K
    lam= 1.36-0.136*(T/100)+0.0057*((T/100)**2)
    return (lam)

Vodivost_d=[]
for teplota in T:
    Vodivost_d.append([Tepelna_vodivost1(teplota)])
"Vykreslení"
print (Vodivost_d)
plt.plot(T,Vodivost_d,'g:',linewidth=4,label='Dolní mez')

# #####
# #####
plt.xlabel('Teplota [ °C ]')
plt.ylabel('Součinitel tepelné vodivosti [ W / mK ]')
plt.legend()
```

Obr. 32 - Součinitel tepelné vodivosti pro beton (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 33):



Obr. 33 - Graf součinitele tepelné vodivosti betonu v závislosti na teplotě (Vytvořeno v prostředí Python podle [5])

2.2.4 Derivace součinitele tepelné vodivosti:

Vstupy (Obr. 34):

```

*****Derivace tepelné vodivosti*****

'''Beton, Horní mez'''
def Derivace_Tepelne_vodivosti0 (T):
    dlam= -0.2451*(1/100)+2*0.0107*T/(100**2)
    return (dlam)

DerVodivost_h=[]
for teplota in T:
    DerVodivost_h.append([Derivace_Tepelne_vodivosti0(teplota)])
"Vykreslení"
print (DerVodivost_h)
plt.plot(T,DerVodivost_h,'r-',linewidth=4,label='Horní mez')

#####
#####

'''Beton, Dolní mez'''
def Derivace_Tepelne_vodivosti1 (T):
    dlam= -0.136/100+2*0.0057*T/(100**2)
    return (dlam)

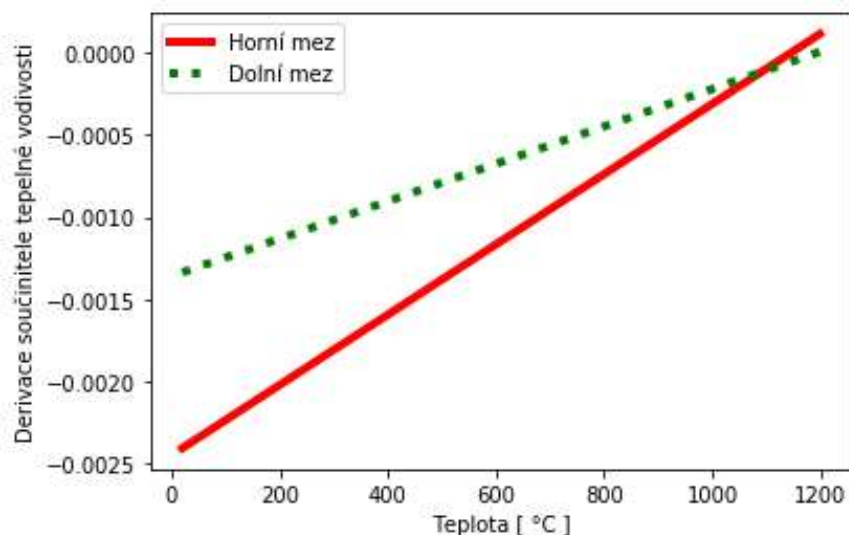
DerVodivost_d=[]
for teplota in T:
    DerVodivost_d.append([Derivace_Tepelne_vodivosti1(teplota)])
"Vykreslení"
print (DerVodivost_d)
plt.plot(T,DerVodivost_d,'g:',linewidth=4,label='Dolní mez')

plt.xlabel('Teplota [ °C ]')
plt.ylabel('Derivace součinitele tepelné vodivosti')
plt.legend()

```

Obr. 34- Derivace součinitele tepelné vodivosti pro beton (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 35):



Obr. 35- Graf derivace součinitele tepelné vodivosti betonu v závislosti na teplotě (Vytvořeno v prostředí Python podle [5])

2.3 Modelování 1D sdílení tepla pro betonové konstrukce

Zde uvedu celé modelování 1D sdílení tepla v konstrukci v programovacím jazyce Python. Uvedu zde také pár příkazů, které budu hodně používat, a vysvětlím, co znamenají.

2.3.1 Používané příkazy

Mezi nejdůležitější příkazy patří příkaz listu. List se značí hranatými závorkami a pro lepší vysvětlení uvedu příklad z mého kódu.

- Teploty = [[T_0] * (nx + 1)]

Tento příkaz vezme teploty T_0, tedy 20°C a napíše je (nx+1)krát za sebou. Tím vytváří první řádek tabulky, který vypadá následovně: [20, 20, 20, 20, 20]. (viz Obr. 36)

Dále se bude postupovat po jednotlivých časových krocích a budou se dopočítávat teploty v konkrétním čase a konkrétních vzdálenostech. Tyto teploty se vždy přidají jako další list. Nakonec se vytvoří jeden velký list, který bude obsahovat (nt + 1) „podlistů“. Pro lepší přehlednost ukážu příklad teplot na obrázku níže (Obr. 36).

Celkový list pak bude vypadat následovně: [[20, 20, 20, ..., 20], [27.5, 20, 20, ..., 20], [33.7, 22.5, 20, ..., 20], [41.1, 25.5, 20.9, ..., 20], ..., [450.9, 352.4, 288.3, ..., 230.3]].

Z tohoto listu lze vykreslit průběh teplot v jakémkoli čase.

Pozn.: Programovací jazyk používá desetinné tečky, čárky pak používá na oddělení jednotlivých čísel.

		tloušťka konstrukce ->								
uzel		0	1	2	3	.	.	.	nx	
		x = 0 m	x = 1*Δx m	x = 2*Δx m	x = 3*Δx m	.	.	.	x = nx*Δx = L m	
krok	0 t = 0 s	T0	T0	T0	T0				T0	[20, 20, 20, ..., 20, 20] T0
	1 t = Δt (5 s)									[27.5, 20, 20, ..., 20, 20] T1
	2 t = 2*Δt (10 s)									[33.7, 22.5, 20, ..., 20, 20] T2
	3 t = 3*Δt (15 s)									[41.1, 25.5, 20.9, ..., 20, 20] T3
	nt t = tlim = nt*Δt (100 s)									[450.9, 352.4, 288.3, ..., 230.3] Tn

Obr. 36 - Pomocná tabulka pro názornou ukázkou principu příkazů

Dalším důležitým příkazem, který jsem již používala pro vytváření materiálových charakteristik, je příkaz na Obr. 37:

```
if T >=20 and T<115:
    ro = ro20
```

Obr. 37 - Příkaz pro proměnnou teplotu - vytvořeno v prostředí Python

Tento příkaz znamená, že pokud je zadané T v rozmezí 20°C až 115°C , pak se ro rovná $ro20$. Tento příkaz umí porovnávat i čísla s desetinnou čárkou (např. pokud bych měla teplotu 80.55 , tak tento příkaz zvládne zařadit tuto teplotu do správného intervalu). Existuje i jiný zápis – viz Obr. 38, který říká to stejné, jen zvládá posoudit pouze celá čísla, což v tomto případě nemohu použít.

```
if T in range (20, 115):
    ro = ro20
```

Obr. 38 - Příkaz pro proměnnou teplotu 2 - vytvořeno v prostředí Python

2.3.2 Kód pro sdílení tepla

Zde uvedu svůj kód z programovacího jazyka python, rozdělený do jednotlivých kroků, ke kterým přidám komentář pro lepší orientaci v tomto kódu. Pro tvorbu tohoto kódu jsem používala rovnice uvedené výše v této práci.

- Nejprve se musí do kódu načíst všechny potřebné knihovny a “pod výpočty” (viz Obr. 39).
 - o V mém případě se jedná o knihovnu matplotlib, která mi umožňuje vykreslení výsledků a také knihovna numpy, která mi vypisuje jednotlivé kroky.
 - o Dále si načtu materiálové charakteristiky a křivky, které jsem si, pro lepší přehlednost, modelovala zvlášť v samostatném souboru.

```
import matplotlib.pyplot as plt
import numpy as np

'''Načtení materiálových charakteristik a křivek'''
import mat_charakteristiky_a_krivky as MT
```

Obr. 39 - Načtení knihoven a pomocných souborů (Vytvořeno v prostředí Python)

- Dále potřebuji zadat vstupní údaje (viz Obr. 40).
 - o Jedná se o tloušťku konstrukce.
 - o Prostorovou a časovou mřížku (viz kapitola 1.5.3 Časová a prostorová diskretizace) vč. omezení délky časového kroku. U omezení časového kroku jsem dala tisknout poznámku, aby se vědělo, že je krok dostatečně malý. Tato poznámka bude vidět na začátku výstupu z kódu.
 - o A materiálové vlastnosti. Zde jsem si připravila řádky pro volbu materiálových vlastností. Např. u Měrného tepla se zde vybere, jestli použijeme suchý beton, nebo beton s vlhkostí a podle toho se u c_{px} změní číslo, buď 0 pro suchý beton, nebo 1,5 nebo 3 pro beton s vlhkostí.

```

'''1D vedeni tepla'''

'''VSTUPNI UDAJE'''

'''Geometrie'''
L = 0.02 # m, tloustka konstrukce

'''Prostorova mrazka'''
dx = 0.005 # m, delka vzdalenostniho kroku
nx = int(L/dx) # pocer kroku
x = np.linspace(0,L,nx+1) # jednotlivé kroky

'''Casova mrazka'''
t_lim = 600 # s, konecny cas
dt = 10 # s, delka casoveho kroku
nt = int(t_lim/dt) # pocer kroku v case

'''Materialove vlastnosti'''
#### Hustota [kg/m3] ####
ro0 = 2350 # Hustota pri 20 stupnich (2350)
#### Měrné teplo [J/kg*K] ####
cpx = 1.5 # 0 pro suchy beton, 1.5 pro 1.5% vlhkost, 3 pro 3% vlkost
#### Tepelna vodivost [W/m*K] ####
lamdax = 0 # 0 pro horni mez, 1 pro spodni mez
#### Derivace tepelne vodivosti #####
lamdaderx = 0 # 0 pro horni mez, 1 pro spodni mez

'''omezení délky časového kroku'''
lam = 1.9514 # 1,951408 horní mez, 1,333028 spodní mez
c = 900
D_T = lam/(ro0*c)
t_crit = (dx**2)/(2*D_T)

if t_crit < dt:
    print("error - zmenčit časový krok", t_crit)
else:
    print("Super! Časový krok je dostratečně malý, jdeme na výpočet.")

```

Obr. 40 - Vstupní údaje (Vytvořeno v prostředí Python)

- Dále budu potřebovat počáteční a okrajové podmínky (viz Obr. 41).
 - o U okrajových podmínek záleží hlavně na tom, jestli je požár z jedné strany nebo ze dvou stran. V tomto případě uvažují s požárem z jedné stany.

```

'''Pocatecni podminka'''
T_0 = 20 # °C

'''Okrajova podminka obecne'''
SB = 5.67*(10**-8) #StefanBoltzmanova konstanta

#-----
'''Okrajove podminky'''
Pozar = 2 # 1 = pozar z leva, 2 = pozar z obou stran, 3= pozar z prava
# pozar z leva -----
if Pozar == 1:
    alfal = 25 #(25-normova; 35-parametricka; 50-uhlovod.)
    epsL = 0.7 # emisivita
    # TL se bude brát jako teplota pozaru
    alfaP = 4 # alfa = 4, eps = 0,7 / alfa = 9, eps = 0 / alfa = 0, eps = 0
    epsP = 0.7 # emisivita
    TP = T_0
# pozar z obou stran -----
elif Pozar == 2:
    alfal = 25 #(25-normova; 35-parametricka; 50-uhlovod.)
    epsL = 0.7 # emisivita
    # TL se bude brát jako teplota pozaru
    alfaP = alfal
    epsP = epsL
    # TP se bude brát jako teplota pozaru
# pozar z prava -----
elif Pozar == 3:
    alfal = 4 # alfa = 4, eps = 0,7 / alfa = 9, eps = 0 / alfa = 0, eps = 0
    epsL = 0.7 # emisivita
    TL = T_0
    alfaP = 25 #(25-normova; 35-parametricka; 50-uhlovod.)
    epsP = 0.7 # emisivita
    # TP se bude brát jako teplota pozaru
#-----

```

Obr. 41 - Okrajové a počáteční podmínky (Vytvořeno v prostředí Python)

- Teď si připravím teploty v prvním řádku a počet časových a vzdálenostních kroků (Obr. 42).

```
'''Teploty v prvním řádku'''
Teploty = [[T_0]*(nx+1)]

times = np.arange(dt,t_lim+dt,dt)
# print(times)
uzly = np.arange(0,nx+1,1)
# print(uzly)
```

Obr. 42 - Teploty pro první řádek (Vytvořeno v prostředí Python)

- Následují vstupy pro parametrickou křivku (Obr. 43 a Obr. 44)
 - o Zde bude na výběr ze dvou možností, možnost 1 bude zadání pouze 3 údajů a to: faktoru otvorů „O“, součinitele „b“ a návrhové hodnoty hustoty požárního zatížení „qtd“. Druhou možností bude podrobné zadání všech vstupů pro parametrickou křivku, jako jsou rozměry místnosti, rozměry a počty otvorů, materiálů místností, atd..

```
'''Vstupy pro parametrickou křivku (O, b, qtd) zadane primo '''
O = 0.05
b = 1520
qtd = 67
'''Rychlost rozvoje požaru'''
RRP = 2;
# 1 ... pomalý rozvoj
# 2 ... středně rychlý rozvoj
# 3 ... rychlý
```

Obr. 43 - Vstupy pro parametrickou křivku zadané přímo (Vytvořeno v prostředí Python podle [1])

```

'''Vstupy pro parametrickou křivku (0, b, qtd) dopocitane ze zakladnich udaju '''
# Rozměry požárního úseku (šířka, délka, výška) [m]:
a = 5
b = 5
h = 3
#-----
Af = a*b
At = 2*(Af+(a+b)*h)
#-----
# Rozměry otvorů (počet, šířka, výška, vážený průměr výšek oken) [m]:
n_o = 2
a_o = 1.5
b_o = 2.5
heq = 2.5
#-----
Av = n_o*a_o*b_o
O = (Av/At)*np.sqrt(heq)
#-----
# Ohraničující konstrukce-podlaha:
ro_1 = 2500
c_1 = 1000
la_1 = 1.4
b_1 = np.sqrt(la_1*ro_1*c_1)
# Ohraničující konstrukce-strop:
ro_2 = 2500
c_2 = 1000
la_2 = 1.4
b_2 = np.sqrt(la_2*ro_2*c_2)
# Ohraničující konstrukce-stěny:
ro_3 = 500
c_3 = 1000
la_3 = 0.13
b_3 = np.sqrt(la_3*ro_3*c_3)
#-----
b = ((b_1*a*b)+(b_2*a*b)+(b_3*2*(a+b)*h))/(At-Av)
#-----
# Výpočet qfd:
qfk = 500
m = 0.8
d1 = 1.2
d2 = 1
dn = 1
#-----
qfd = qfk*m*d1*d2*dn
qtd = (qfd*Af)/At

```

Obr. 44- Vstupy pro parametrickou křivku zadané základními údaji (Vytvořeno v prostředí Python podle [1])

- Následuje tvorba kódu. Jedná se o příkaz „for time in times“, který projde jednotlivé časové kroky a v každém spočítá údaje potřebné pro výpočet. V této funkci je pak vepsaná funkce „for uzal in uzly“, která provádí výpočet v každém jednotlivém uzlu v konkrétním čase.
 - o Na Obr. 45 je definice požáru, kde si pomocí čísla volím, podle jaké křivky bude výpočet probíhat. Dále jsou zde připravené listy pro materiálové vlastnosti, které se budou plnit v dalším kroku.

```

for time in times:
    #
    krok = int(np.around(time/dt))
    #
    #-----
    # '''Definice pozaru'''
    Tn = 1 # 1= normová, 2= uhlovodíková,
    #      3= křivka vnějšího požáru, 4= parametrická všechny vstupy
    #      5= parametrická 0,b,qtđ
    if Tn == 1:
        TL = MT.Normova_krivka(time/60)
    elif Tn == 2:
        TL = MT.Uhlovodikova_krivka(time/60)
    elif Tn == 3:
        TL = MT.Krivka_vnejsiho_pozaru(time/60)
    elif Tn == 4:
        TL = MT.ParamTemp2(time/3600,O,b,qtđ, RRP) [0]
    elif Tn == 5:
        TL = MT.ParamTemp(time/3600,O1,b1,qtđ1, RRP) [0]
    if Pozar == 2:
        TP=TL
    if Pozar == 3:
        TP=TL
        TL=0
    #-----
    '''Materialove vlastnosti'''
    ro = []
    cp = []
    lamda = []
    lamdader = []

    for uzal in uzly:

```

Obr. 45 - Tvorba kódu 1. část (Vytvořeno v prostředí Python)

- Pokračování příkazu na Obr. 46 vnořeným příkazem „for uzal in uzly“. V tomto kroku se doplňují listy materiálových vlastností vytvořené v předchozím kroku podle toho, jaké materiálové vlastnosti byly zvoleny (cpx = 0, tedy suchý beton, cpx=1,5, tedy beton s 1,5% vlhkostí, ...)
- Tento „vnořený“ příkaz se věnuje pouze materiálovým vlastnostem. Příkaz „T_krok = []“ už není součástí tohoto příkazu a ten si připravuje listy pro teploty, které se budou doplňovat v dalším kroku.

```

for uzal in uzly:
    Teplota_v_uzlu = Teploty[krok-1][uzal]
    ### Hustota ###
    ro.append(MT.Objemova_hmotnost(Teplota_v_uzlu,ro0))
    ### Merne teplo ###
    if cpx == 0:
        cp.append(MT.Merne_teplo_0(Teplota_v_uzlu))
    elif cpx == 1.5:
        cp.append(MT.Merne_teplo_1(Teplota_v_uzlu))
    elif cpx == 3:
        cp.append(MT.Merne_teplo_3(Teplota_v_uzlu))
    else:
        cp.append(MT.Merne_teplo_0(Teplota_v_uzlu))
    ### Tepelna vodivost ###
    if lamdax == 0:
        lamda.append(MT.Tepelna_vodivost0(Teplota_v_uzlu))
    elif lamdax == 1:
        lamda.append(MT.Tepelna_vodivost1(Teplota_v_uzlu))
    else:
        lamda.append(MT.Tepelna_vodivost0(Teplota_v_uzlu))
    ### Derivace tepelne vodivosti ###
    if lamdaderx == 0:
        lamdader.append(MT.Derivace_Tepelne_vodivosti0(Teplota_v_uzlu))
    elif lamdaderx == 1:
        lamdader.append(MT.Derivace_Tepelne_vodivosti1(Teplota_v_uzlu))
    else:
        lamdader.append(MT.Derivace_Tepelne_vodivosti0(Teplota_v_uzlu))
    #-----
    T_krok = []
    for uzal in uzly:

```

Obr. 46- Tvorba kódu 2. část (Vytvořeno v prostředí Python)

- Na Obr. 47 pokračuje kód druhým „vnořeným“ příkazem, stejném jako v předchozí části a tedy „for uzel in uzly“. Tento příkaz doplňuje teploty do listů podle toho, jestli se nachází na levém nebo pravém okraji nebo v prostřední části.
- Příkaz „for time in times“ zde končí. Je zde možnost vykreslení průběhu rozložení teplot v konstrukci po určitém čase – tedy kód vytvoří video s postupným nárůstem teplot.

```

T_krok = []
for uzel in uzly:
    if uzel == 0:
        #-----
        # Levy uzel
        qL = alfaL*(Teploty[krok - 1][uzel] - TL) + \
            epsL*SB*( (Teploty[krok - 1][uzel] + 273.15)**4 - (TL + 273.15)**4 )
        T_krok.append(Teploty[krok - 1][uzel] + \
            (lamdader[uzel]*dt/(ro[uzel]*cp[uzel]))*(qL/lamda[uzel])**2 + \
            (2*dt*lamda[uzel])/(ro[uzel]*cp[uzel]*dx**2) * \
            (- Teploty[krok - 1][uzel] + Teploty[krok - 1][uzel + 1] - dx*qL/lamda[uzel]))
        #-----
    elif uzel == nx:
        #-----
        # Pravy uzel
        qP = alfaP*(Teploty[krok- 1][uzel]-TP) + \
            epsP*SB*( (Teploty[krok- 1][uzel]+273.15)**4 - (TP+ 273.15)**4)
        T_krok.append(Teploty[krok- 1][uzel] + \
            (lamdader[uzel]*dt/(ro[uzel]*cp[uzel]))*(-qP/lamda[uzel])**2 + \
            (2*dt*lamda[uzel])/(ro[uzel]*cp[uzel]*dx**2) * \
            (-Teploty[krok- 1][uzel] + Teploty[krok- 1][uzel- 1] - dx*qP/lamda[uzel]))
        #-----
    else:
        #-----
        # Vnitřní uzly
        T_krok.append(Teploty[krok- 1][uzel] + \
            (lamdader[uzel]*dt/(ro[uzel]*cp[uzel])) * \
            ((Teploty[krok- 1][uzel+ 1] - Teploty[krok- 1][uzel- 1])/(2*dx))**2 + \
            (dt*lamda[uzel])/(ro[uzel]*cp[uzel]*dx**2) * \
            (Teploty[krok- 1][uzel+1] - 2*Teploty[krok- 1][uzel] + Teploty[krok- 1][uzel- 1]))
        #-----
    Teploty.append(T_krok)
#-----
# plt.plot(x,T_krok)
# # plt.pause(0.01)
# plt.axis([0, L, 0, 800])
#-----

```

Obr. 47- Tvorba kódu 3. část (Vytvořeno v prostředí Python)

- Nakonec už je jen kód k vykreslení grafu (Obr. 48).

```

'''Vykreslení'''
plt.plot(x,T_krok, label= "Průběh teploty")
plt.axis([0, L, 0, 150])
plt.grid(True)
print(Teploty)

plt.xlabel('Tloušťka konstrukce [ m ] ')
plt.ylabel('Teplota [ °C ]')
plt.legend()

```

Obr. 48 - Tvorba kódu - vykreslení (Vytvořeno v prostředí Python)

2.4 Porovnání výstupů

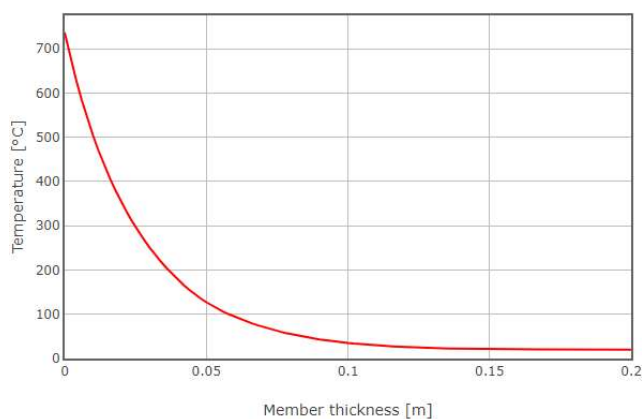
Zde porovnám výstupy z mého kódu s výstupy z programů TeDisOn [1P] a FiDeS [3P] (neboli TempAnalysis). Veškeré vstupní údaje jsou nastaveny stejně pro všechny programy a následné srovnání je provedeno tabulkově v Excelu.

Vstupní údaje:

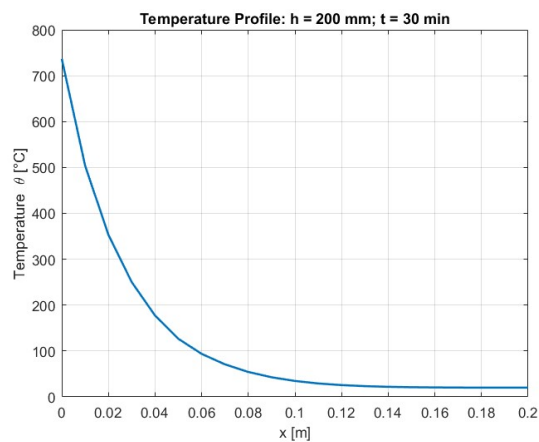
- Tloušťka: 0,2 m
- Doba vystavení požáru: 30 min
- Hustota: 2 400 kg/m³
- Vlhkost betonu: 1,5 %
- Tepelná vodivost: horní mez
- Expozice: 1) požár z 1 strany, 2) požár ze 2 stran

2.4.1 Požár z jedné strany

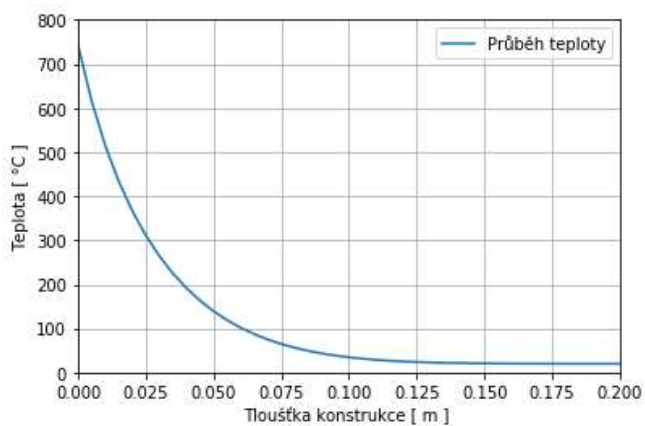
Ukázka výstupu z jednotlivých programů a mého kódu (Obr. 52, Obr. 53 a Obr. 54):



Obr. 52 - TeDisOn - Průběh teploty v konstrukci - požár z jedné strany [1P]

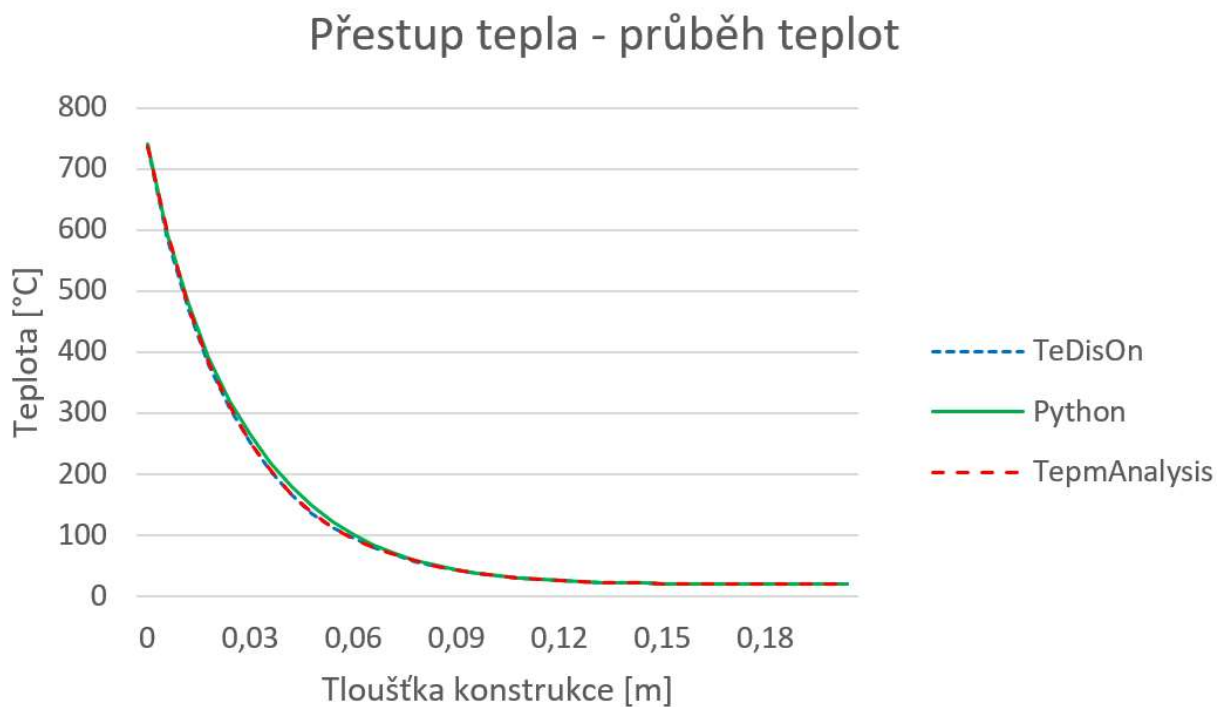


Obr. 53 - FiDeS - Průběh teploty v konstrukci - požár z jedné strany [3P]



Obr. 54 - Python - Průběh teploty v konstrukci - požár z jedné strany

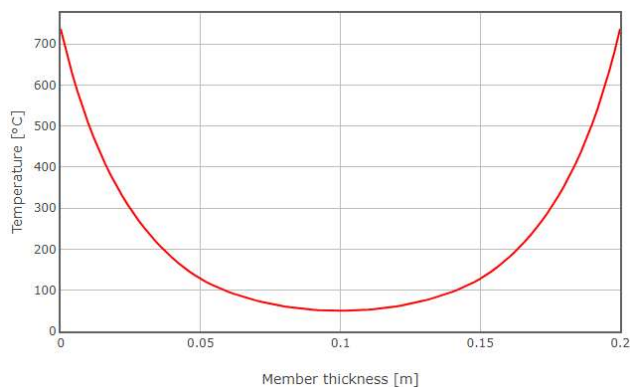
Srovnání jednotlivých křivek (Obr. 55):



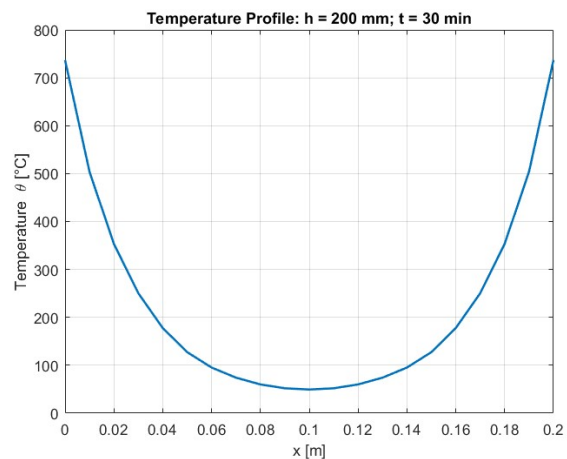
Obr. 55 - Průběh teplot v konstrukci - požár z jedné strany

2.4.2 Požár ze dvou stran

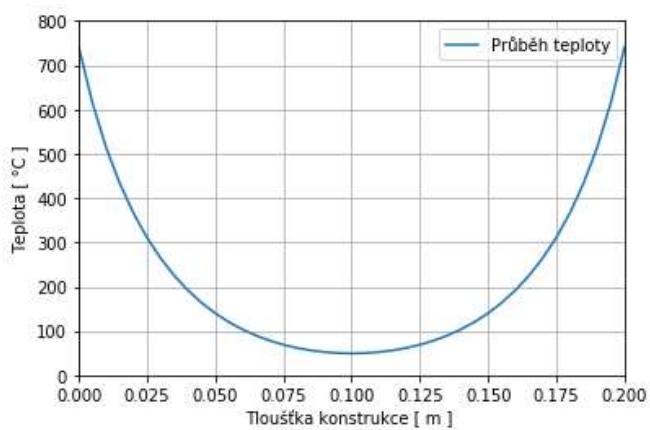
Ukázka výstupu z jednotlivých programů a mého kódu (Obr. 56, Obr. 57 a Obr. 58):



Obr. 56 - TeDisOn - Průběh teploty v konstrukci - požár ze 2 stran [1P]

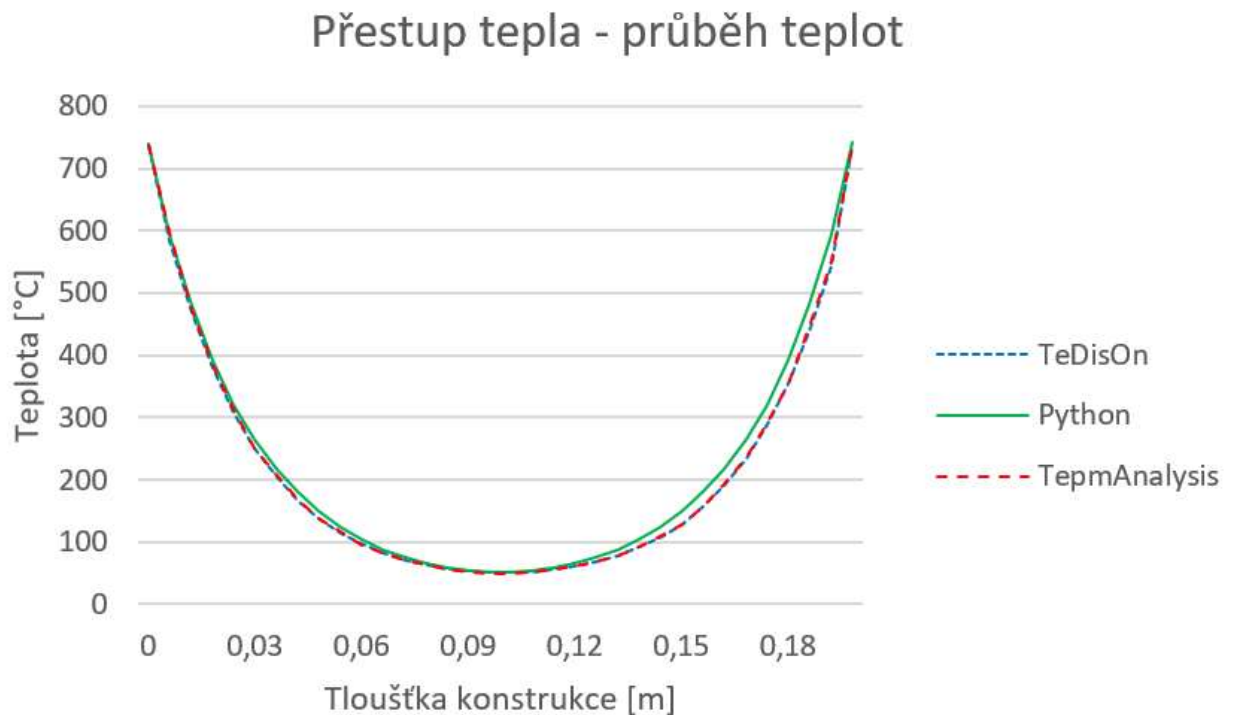


Obr. 57 - FiDeS - Průběh teploty v konstrukci - požár ze 2 stran [3P]



Obr. 58 - Python - Průběh teploty v konstrukci - požár ze 2 stran

Srovnání jednotlivých křivek (Obr. 59):



Obr. 59 - Průběh teplot v konstrukci - požár ze 2 stran

2.4.3 Závěr

Jak je vidět z obrázků výše (Obr. 55 a Obr. 59), průběh křivky mého kódu je shodný s průběhy teplot v programech TeDisOn a FiDeS. Malé rozdíly v křivkách mohou být způsobeny zvolenou délkou kroku (čím menší krok, tím přesnější výpočet, ale také delší výpočetní čas). Tímto srovnáním jsem prokázala funkčnost mého kódu.

Kapitola 3

Praktická část

Praktická část je rozdělena na 3 bloky, přičemž každý je věnován popisu jiné problematiky.

1. První blok se věnuje sestrojení kódu pro 1D modelování přestupu tepla skrz konstrukce.
2. Druhá část se věnuje sestrojení kódu pro výpočet momentu únosnosti pomocí proužkové metody.
3. Třetí blok se věnuje sestrojení kódu pro posouzení požární odolnosti podle metody izotermy 500 °C a zónové metody.

3 Praktická část – 2. blok

Ve 2. bloku praktické části se budu věnovat sestrojení kódu pro výpočet momentu únosnosti pomocí proužkové metody.

Uvedu zde kódy z programovacího jazyka pro jednotlivé součinitele potřebné k výpočtům a také pro výpočty momentů podle proužkové metody.

Dále zde uvedu porovnání výpočtů momentů únosnosti pomocí proužkové metody a výpočtů zjednodušenou metodou.

Na závěr vymodeluji proužkovou metodu pro výpočet při požární situaci a opět porovnáím výsledky.

3.1 Vstupní údaje

Do proužkové metody vstupují pracovní diagramy betonu a oceli, jak za běžné teploty, tak za požáru. Ke všem těmto údajům zde ukážu kód, kde bude vidět, jak jsem pracovní diagramy modelovala a jaké parametry do nich vstupují. Veškeré vzorečky brány z eurokódů 1992-1-1 [11] a 1992-1-2 [5].

3.1.1 Pracovní diagram betonu za běžné teploty

Zde jsem modelovala dva pracovní diagramy, a to Parabolicko-rektangulární a Bilineární. Oba pracovní diagramy jsou pro běžný beton, do třídy C50/60).

Vstup (Obr. 60 a Obr. 61):

```
fck = 30
fcd = fck/1.5

ec = np.arange(0,3.5+0.001,0.001)

def Parabolicko_rektangularni (ec,fcd):
    n = 2      # dle tab. 3.1 v 1992-1-1 (beton do tr.50/60)
    ec2 = 2    # dle tab. 3.1 v 1992-1-1 (beton do tr.50/60)
    ecu2 = 3.5 # dle tab. 3.1 v 1992-1-1 (beton do tr.50/60)
    if ec > 0 and ec <= ec2:
        sigm = fcd*(1-(1-ec/ec2)**n)
    elif ec > ec2 and ec <= ecu2:
        sigm = fcd
    else:
        sigm = 0
    return (sigm)

parabolicko_rektangularni = []
for pretvoreni in ec:
    parabolicko_rektangularni.append([Parabolicko_rektangularni(pretvoreni,fcd)])

"Vykreslení"
print (parabolicko_rektangularni)
plt.plot(ec,parabolicko_rektangularni,'b',linewidth=4,label = "Parabolicko rektangulární")
```

Obr. 60 - Parabolicko_rektangulární pracovní diagram betonu ((Vytvořeno v prostředí Python podle [11]))


```

def Bilinearni (ec,fcd):
    ec3 = 1.75      # dle tab. 3.1 v 1992-1-1 (beton do tr.50/60)
    ecu3 = 3.5     # dle tab. 3.1 v 1992-1-1 (beton do tr.50/60)
    if ec > 0 and ec <= ec3:
        sigm = fcd*ec/ec3
    elif ec > ec3 and ec <= ecu3:
        sigm = fcd
    else:
        sigm = 0
    return (sigm)

bilinearni = []
for pretvoreni in ec:
    bilinearni.append(Bilinearni(pretvoreni,fcd))

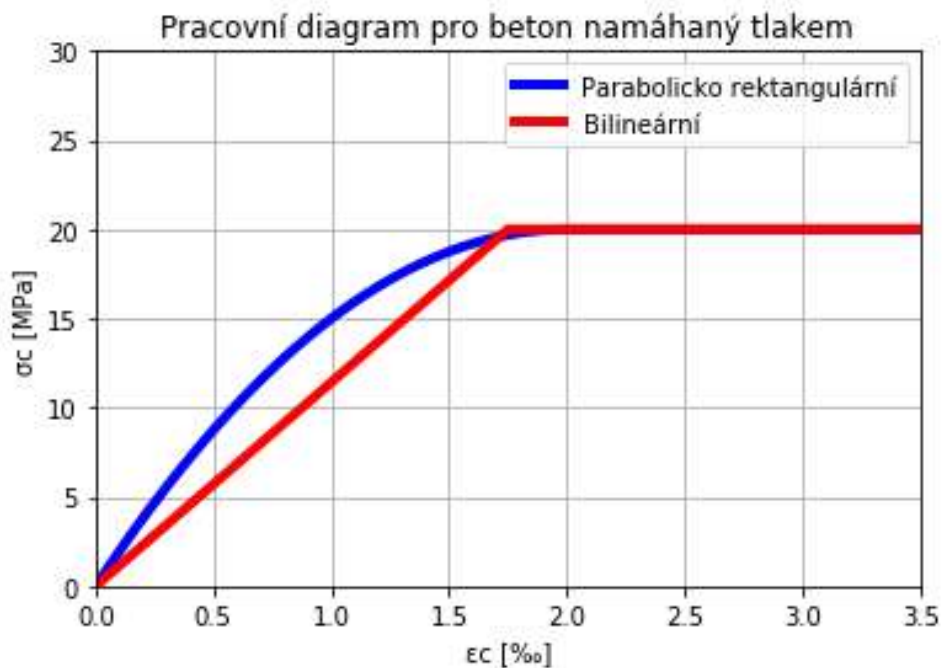
"Vykreslení"
print (parabolicko_rektangularni)
plt.plot(ec,bilinearni,'r',linewidth=4, label = "Bilineární")

plt.grid(True)
plt.title('Pracovní diagram pro beton namáhaný tlakem')
plt.axis([0, 3.5, 0, fcd+10])
plt.xlabel('\u03b5c [%o]')
plt.ylabel('\u03c3c [MPa]')
plt.legend()

```

Obr. 61 - Bilineární pracovní diagram betonu (Vytvořeno v prostředí Python podle [11])

Výstup (Obr. 62):



Obr. 62 - Pracovní diagram betonu - graf (Vytvořeno v prostředí Python podle [11])

3.1.2 Pracovní diagram betonu za požární situace

Do pracovního diagramu vstupuje součinitel k_s v závislosti na teplotě a také napětí závislé na teplotě. Tyto parametry uvedu níže.

Funkce (Obr. 63):

```
def Pozar_diagram (ec,T,fck,k):
    if k == 1:
        kc = Soucinitel_k1(T)
        fc_temp = kc * fck
    else:
        kc = Soucinitel_k2(T)
        fc_temp = kc * fck
    ec1_temp = ec1_temper(T)
    ecu1_temp = ecu1_temper(T)
    if ec > 0 and ec < ec1_temp:
        sigm = (3*ec*fc_temp)/(ec1_temp*(2+(ec/ec1_temp)**3))
    elif ec >= ec1_temp and ec < ecu1_temp:
        sigm = fc_temp*((ecu1_temp-ec)/(ecu1_temp-ec1_temp))
    else:
        sigm = 0
    return (sigm)
```

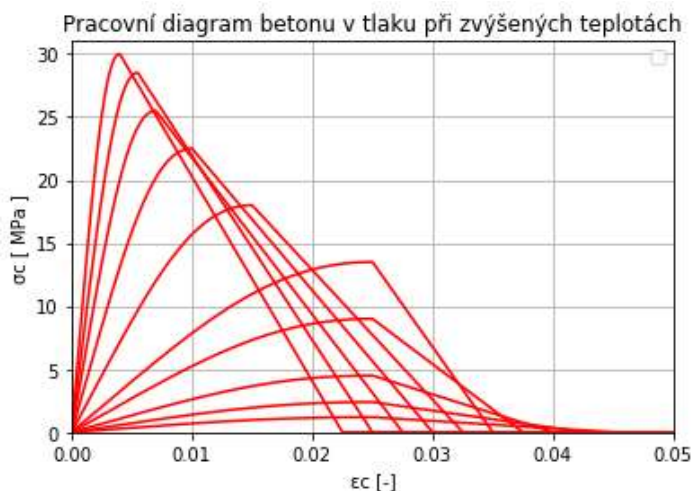
Obr. 63 - Pracovní diagram betonu za požáru - funkce (Vytvořeno v prostředí Python podle [5])

Vykreslení (Obr. 64):

```
teploty = [100,200,300,400,500,600,700,800,900,1000]
for teplota in teploty:
    eps = np.arange(-0.002,0.05,0.00001)
    sigma = []
    for i in range(0,len(eps)):
        sigma.append(Pozar_diagram (eps[i],teplota,30e6,1))
    plt.plot(eps,sigma,'r')
    plt.axis([0, 0.05, 0, 30e6])
    # plt.pause(0.01)
plt.grid(True)
plt.title("Pracovní diagram betonu v tlaku při zvýšených teplotách")
plt.xlabel('\u03B5c [-]')
plt.ylabel('\u03C3c [ MPa ]')
plt.legend()
```

Obr. 64 - Pracovní diagram betonu za požáru – kód pro vykreslení (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 65):



Obr. 65 - Pracovní diagram betonu za požáru - graf (Vytvořeno v prostředí Python podle [5])

3.1.2.1 Součinitel ks

Součinitel slouží pro redukci charakteristické pevnosti betonu, která závisí na teplotě betonu. Součinitel se uvažuje podle druhu kameniva buď pro beton s křemičitým nebo vápencovým kamenivem.

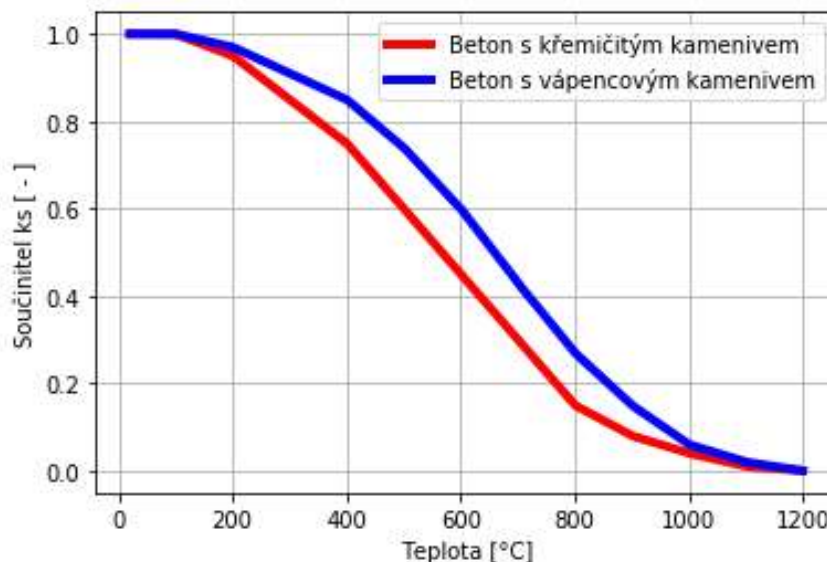
Vstup (Obr. 66):

```
def Soucinitel_k1 (T):
    if T >=20 and T<100:
        ks = 1
    elif T >=100 and T<200:
        ks = 0.95-0.05*((T-200)/100)
    elif T >=200 and T<400:
        ks = 0.75-0.2*((T-400)/200)
    elif T >=400 and T<800:
        ks = 0.15-0.6*((T-800)/400)
    elif T >=800 and T<900:
        ks = 0.08-0.07*((T-900)/100)
    elif T >=900 and T<1000:
        ks = 0.04-0.04*((T-1000)/100)
    elif T >=1000 and T<1100:
        ks = 0.01-0.03*((T-1100)/100)
    elif T >=1100 and T<=1200:
        ks = 0.01*((1200-T)/100)
    else:
        ks = 0
    return (ks)
```

```
def Soucinitel_k2 (T):
    if T >=20 and T<100:
        ks = 1
    elif T >=100 and T<200:
        ks = 0.97-0.03*((T-200)/100)
    elif T >=200 and T<400:
        ks = 0.85-0.12*((T-400)/200)
    elif T >=400 and T<500:
        ks = 0.74-0.11*((T-500)/100)
    elif T >=500 and T<600:
        ks = 0.6-0.14*((T-600)/100)
    elif T >=600 and T<700:
        ks = 0.43-0.17*((T-700)/100)
    elif T >=700 and T<800:
        ks = 0.27-0.16*((T-800)/100)
    elif T >=800 and T<900:
        ks = 0.15-0.12*((T-900)/100)
    elif T >=900 and T<1000:
        ks = 0.06-0.09*((T-1000)/100)
    elif T >=1000 and T<1100:
        ks = 0.02-0.04*((T-1100)/100)
    elif T >=1100 and T<=1200:
        ks = 0.02*((1200-T)/100)
    else:
        ks = 0
    return (ks)
```

Obr. 66 - Součinitel ks pro beton s křemičitým kamenivem (k1)/ s vápencovým kamenivem (k2)
(Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 67):



Obr. 67 - Součinitel ks pro redukci pevnosti betonu – graf závislosti ks na teplotě (Vytvořeno v prostředí Python podle [5])

3.1.2.2 Přetvoření betonu

Přetvoření betonu také závisí na teplotě, zde jsem namodelovala rovnice, které jsem vytvořila dle tabulky 3.1 v eurokódu 1992-1-2 [5].

Vstupy (Obr. 68 a Obr. 69):

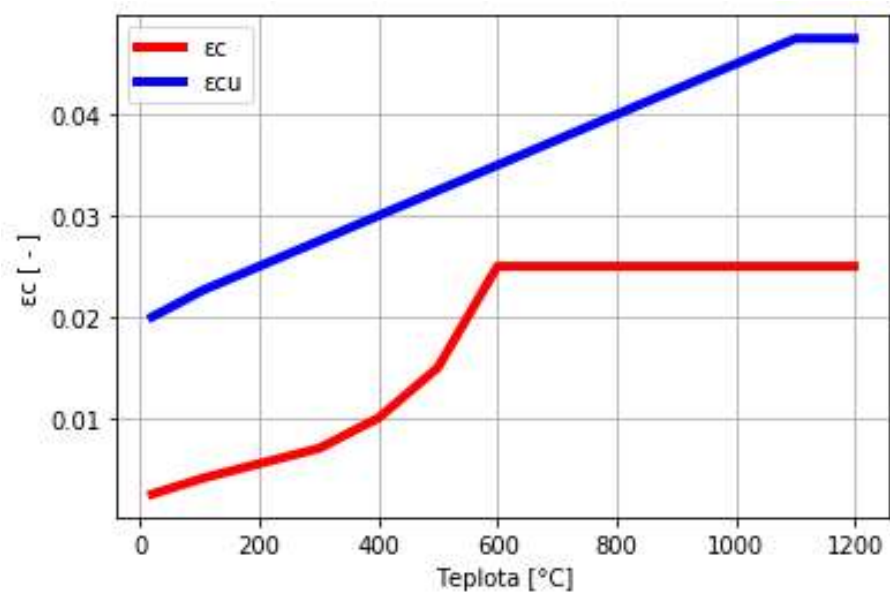
```
def ec1_temper (T):
    if T >=20 and T<100:
        ec1 = 0.0040+0.0015*((T-100)/80)
    elif T >=100 and T<200:
        ec1 = 0.0055+0.0015*((T-200)/100)
    elif T >=200 and T<300:
        ec1 = 0.0070+0.0015*((T-300)/100)
    elif T >=300 and T<400:
        ec1 = 0.0100+0.003*((T-400)/100)
    elif T >=400 and T<500:
        ec1 = 0.0150+0.005*((T-500)/100)
    elif T >=500 and T<600:
        ec1 = 0.0250+0.01*((T-600)/100)
    elif T >=600 and T<=1200:
        ec1 = 0.0250
    else:
        ec1 = 0
    return (ec1)
```

Obr. 68 - Přetvoření betonu ϵ_c (Vytvořeno v prostředí Python podle[5])

```
def ecui_temper (T):
    if T >=20 and T<100:
        ecui = 0.0225+0.0025*((T-100)/80)
    elif T >=100 and T<200:
        ecui = 0.0250+0.0025*((T-200)/100)
    elif T >=200 and T<300:
        ecui = 0.0275+0.0025*((T-300)/100)
    elif T >=300 and T<400:
        ecui = 0.0300+0.0025*((T-400)/100)
    elif T >=400 and T<500:
        ecui = 0.0325+0.0025*((T-500)/100)
    elif T >=500 and T<600:
        ecui = 0.0350+0.0025*((T-600)/100)
    elif T >=600 and T<700:
        ecui = 0.0375+0.0025*((T-700)/100)
    elif T >=700 and T<800:
        ecui = 0.0400+0.0025*((T-800)/100)
    elif T >=800 and T<900:
        ecui = 0.0425+0.0025*((T-900)/100)
    elif T >=900 and T<1000:
        ecui = 0.0450+0.0025*((T-1000)/100)
    elif T >=1000 and T<1100:
        ecui = 0.0475+0.0025*((T-1100)/100)
    elif T >=1100 and T<=1200:
        ecui = 0
    else:
        ecui = 0
    return (ecui)
```

Obr. 69 – Mezní přetvoření betonu ϵ_{cu} (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 70):



Obr. 70 - Graf závislosti přetvoření na teplotě (Vytvořeno v prostředí Python podle [5])

3.1.3 Pracovní diagram oceli za běžné teploty

Při modelování pracovního diagramu oceli jsem použila idealizovaný diagram, kdy je horní větev vodorovná, bez nutnosti omezení poměrného přetvoření.

Vstup (Obr. 71):

```
fyk = 500
fyd = fyk/1.15
Es = 200e3
eu = np.arange(0,5.5+0.001,0.00001)

def PD_Ocel (eu,fyd,Es):
    eu1= 1000*fyd/Es # přetvoření na mezi kluzu
    if eu > 0 and eu <= eu1:
        sigm = fyd*eu/eu1
    elif eu > eu1:
        sigm = fyd
    else:
        sigm = 0
    return (sigm)

pd_ocel = []
for pretvoreni in eu:
    pd_ocel.append([PD_Ocel(pretvoreni,fyd,Es)])

"Vykreslení"
plt.plot(eu,pd_ocel,'b',linewidth=4)
```

Obr. 71 - Pracovní diagram oceli (Vytvořeno v prostředí Python podle [11])

Výstup (Obr. 72):



Obr. 72 - Pracovní diagram oceli - graf (Vytvořeno v prostředí Python podle [11])

3.1.4 Pracovní diagram oceli za požáru

Pracovní diagram oceli za požáru je definován třemi parametry – sklonem v lineárně pružné oblasti $E_{s,\theta}$, mezí úměrnosti $f_{sp,\theta}$ a maximálním napětím $f_{sy,\theta}$. Postupně zde ukážu kódy ke všem parametrům potřebným pro výpočet.

Vstup pro pracovní diagram (Obr. 73):

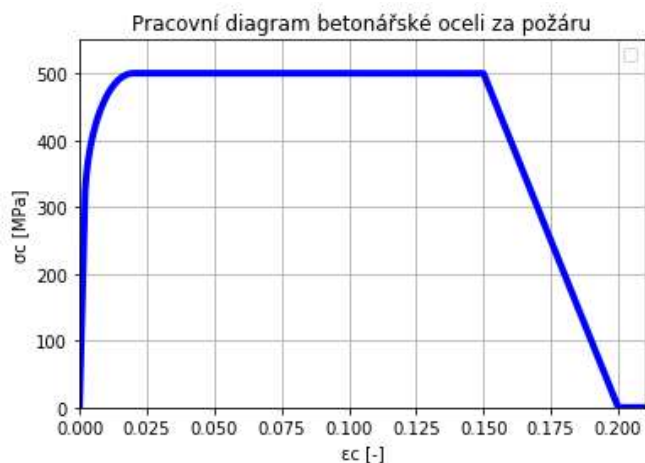
```
def PD_Ocel_pozar (eps,T,fyk,k,kk,kE,Es):
    if k == 1:
        ks = Soucinitel_k1(T)
        fsy = ks * fyk
    elif k == 2:
        ks = Soucinitel_k2(T)
        fsy = ks * fyk
    else:
        ks = Soucinitel_k3(T)
        fsy = ks * fyk
    if kk == 1:
        kks = Soucinitel_kk1(T)
        fsp = kks*fyk
    else:
        kks = Soucinitel_kk2(T)
        fsp = kks*fyk
    if kE == 1:
        ke = Soucinitel_E1(T)
        Es_temp = ke*Es
    else:
        ke = Soucinitel_E2(T)
        Es_temp = ke*Es

    esp = fsp/Es_temp
    esy = 0.02
    est = 0.15
    esu = 0.20
    c = ((fsy-fsp)**2)/((esy-esp)*Es_temp-2*(fsy-fsp))
    a = np.sqrt((esy-esp)*(esy-esp+c/Es_temp))
    b = np.sqrt(c*(esy-esp)*Es_temp+c**2)

    if eps > 0 and eps <= esp:
        sigm = eps*Es_temp
    elif eps > esp and eps <= esy:
        sigm = fsp - c + (b/a)*(a**2-(esy-eps)**2)**0.5
    elif eps > esy and eps <= est:
        sigm = fsy
    elif eps > est and eps <= esu:
        sigm = fsy*(1-(eps-est)/(esu-est))
    else:
        sigm = 0
    return (sigm)
```

Obr. 73 - Pracovní diagram oceli za požáru (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 74):



Obr. 74 - Pracovní diagram oceli za požáru – graf (Vytvořeno v prostředí Python podle [5])

3.1.4.1 Součinitel k_s

Součinitel slouží pro redukci charakteristické pevnosti oceli, která závisí na teplotě betonu. Součinitel se uvažuje podle druhu výztuže buď pro tahovou výztuž válcovanou za tepla (křivka 1) nebo za studena (křivka 2), při poměrném přetvoření nad 2 % nebo tlakovou a tahovou výztuž při poměrném přetvoření do 2 %.

Vstupy (Obr. 75 a Obr. 76):

```
def Soucinitel_k1 (T):
    if T >=20 and T<400:
        ks = 1
    elif T >=400 and T<500:
        ks = 0.78-0.22*((T-500)/100)
    elif T >=500 and T<600:
        ks = 0.47-0.31*((T-600)/100)
    elif T >=600 and T<700:
        ks = 0.23-0.24*((T-700)/100)
    elif T >=700 and T<800:
        ks = 0.11-0.12*((T-800)/100)
    elif T >=800 and T<900:
        ks = 0.06-0.05*((T-900)/100)
    elif T >=900 and T<=1200:
        ks = 0.06*((1200-T)/300)
    else:
        ks = 0
    return (ks)

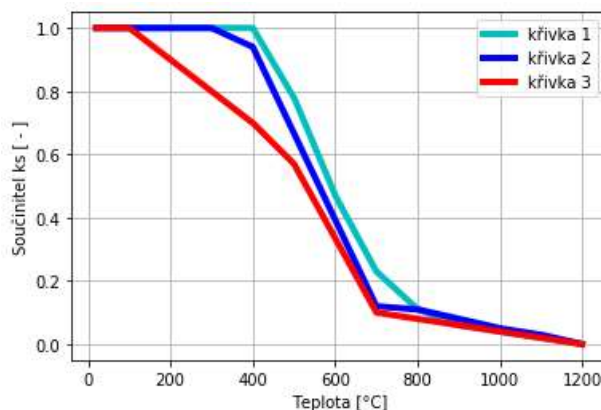
def Soucinitel_k2 (T):
    if T >=20 and T<300:
        ks = 1
    elif T >=300 and T<400:
        ks = 0.94-0.06*((T-400)/100)
    elif T >=400 and T<700:
        ks = 0.12-0.82*((T-700)/300)
    elif T >=700 and T<800:
        ks = 0.11-0.01*((T-800)/100)
    elif T >=800 and T<1000:
        ks = 0.05-0.06*((T-1000)/200)
    elif T >=1000 and T<1100:
        ks = 0.03-0.02*((T-1100)/100)
    elif T >=1100 and T<=1200:
        ks = 0.03*((1200-T)/100)
    else:
        ks = 0
    return (ks)
```

Obr. 75 - Součinitel k_s pro tahovou výztuž válcovanou za tepla (k1) a za studena (k2) (Vytvořeno v prostředí Python podle [5])

```
def Soucinitel_k3 (T):
    if T >=20 and T<100:
        ks = 1
    elif T >=100 and T<400:
        ks = 0.7-0.3*((T-400)/300)
    elif T >=400 and T<500:
        ks = 0.57-0.13*((T-500)/100)
    elif T >=500 and T<700:
        ks = 0.1-0.47*((T-700)/200)
    elif T >=700 and T<=1200:
        ks = 0.1*((1200-T)/500)
    else:
        ks = 0
    return (ks)
```

Obr. 76 - Součinitel k_s pro tlakovou a tahovou výztuž (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 77):



Obr. 77 - Součinitel k_s pro ocel – graf závislosti k_s na teplotě (Vytvořeno v prostředí Python podle [5])

3.1.4.2 Poměr meze úměrnosti a pevnosti oceli

Pro lepší funkčnost kódu jsem přepsala data z tabulky 3.2a z eurokódu 1992-1-2 [5] do vzorců, které vytvářejí křivku. Kód pak nebude muset zaokrouhlovat čísla, ale vezme rovnou číslo z křivky. Tento poměr je také rozdělen podle toho, jestli se jedná o výztuž taženou za tepla, nebo taženou za studena.

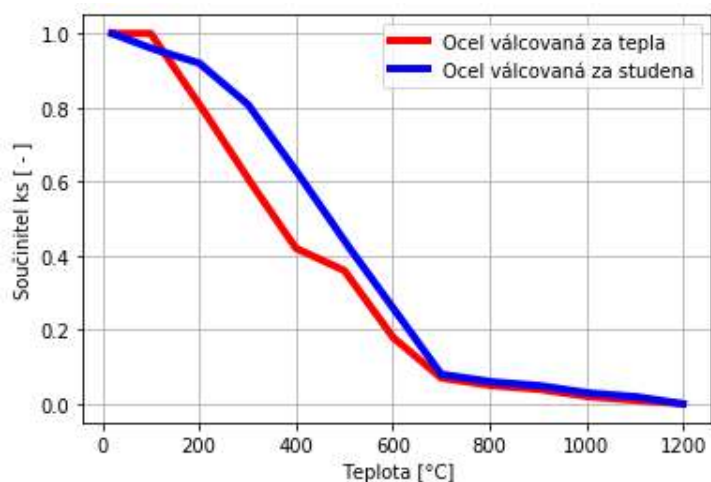
Vstupy (Obr. 78):

```
def Soucinitel_kk1 (T):
    if T >=20 and T<100:
        kk = 1
    elif T >=100 and T<200:
        kk = 0.81-0.19*((T-200)/100)
    elif T >=200 and T<300:
        kk = 0.61-0.20*((T-300)/100)
    elif T >=300 and T<400:
        kk = 0.42-0.19*((T-400)/100)
    elif T >=400 and T<500:
        kk = 0.36-0.06*((T-500)/100)
    elif T >=500 and T<600:
        kk = 0.18-0.18*((T-600)/100)
    elif T >=600 and T<700:
        kk = 0.07-0.11*((T-700)/100)
    elif T >=700 and T<800:
        kk = 0.05-0.02*((T-800)/100)
    elif T >=800 and T<900:
        kk = 0.04-0.01*((T-900)/100)
    elif T >=900 and T<1000:
        kk = 0.02-0.02*((T-1000)/100)
    elif T >=1000 and T<1100:
        kk = 0.01-0.01*((T-1100)/100)
    elif T >=1100 and T<=1200:
        kk = 0.01*((1200-T)/100)
    else:
        kk = 0
    return (kk)
```

```
def Soucinitel_kk2 (T):
    if T == 20:
        kk = 1
    elif T >20 and T<100:
        kk = 0.96-0.04*((T-100)/80)
    elif T >=100 and T<200:
        kk = 0.92-0.04*((T-200)/100)
    elif T >=200 and T<300:
        kk = 0.81-0.11*((T-300)/100)
    elif T >=300 and T<400:
        kk = 0.63-0.18*((T-400)/100)
    elif T >=400 and T<500:
        kk = 0.44-0.19*((T-500)/100)
    elif T >=500 and T<600:
        kk = 0.26-0.18*((T-600)/100)
    elif T >=600 and T<700:
        kk = 0.08-0.18*((T-700)/100)
    elif T >=700 and T<800:
        kk = 0.06-0.02*((T-800)/100)
    elif T >=800 and T<900:
        kk = 0.05-0.01*((T-900)/100)
    elif T >=900 and T<1000:
        kk = 0.03-0.02*((T-1000)/100)
    elif T >=1000 and T<1100:
        kk = 0.02-0.01*((T-1100)/100)
    elif T >=1100 and T<=1200:
        kk = 0.02*((1200-T)/100)
    else:
        kk = 0
    return (kk)
```

Obr. 78 - Poměr meze úměrnosti a pevnosti oceli (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 79):



Obr. 79 - Graf poměru meze únosnosti a pevnosti oceli v závislosti na teplotě (Vytvořeno v prostředí Python podle [5])

3.1.4.3 Poměr sklonu v lineárně pružné oblasti s modulem pružnosti

Zde jsem také pro lepší funkčnost kódu přepsala data z tabulky 3.2a z eurokódu 1992-1-2 [5] do vzorců, které vytvářejí křivku. Kód pak nebude muset zaokrouhlovat čísla, ale vezme rovnou číslo z křivky. Tento poměr je také rozdělen podle toho, jestli se jedná o výztuž taženou za tepla nebo taženou za studena.

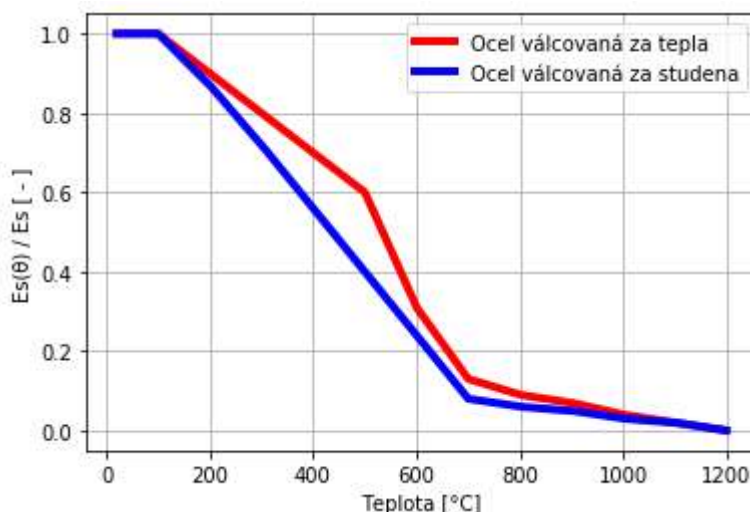
Vstupy (Obr. 80):

```
def Soucinitel_E1 (T):
    if T >=20 and T<100:
        E = 1
    elif T >=100 and T<200:
        E = 0.9-0.1*((T-200)/100)
    elif T >=200 and T<300:
        E = 0.8-0.1*((T-300)/100)
    elif T >=300 and T<400:
        E = 0.7-0.1*((T-400)/100)
    elif T >=400 and T<500:
        E = 0.6-0.1*((T-500)/100)
    elif T >=500 and T<600:
        E = 0.31-0.29*((T-600)/100)
    elif T >=600 and T<700:
        E = 0.13-0.18*((T-700)/100)
    elif T >=700 and T<800:
        E = 0.09-0.04*((T-800)/100)
    elif T >=800 and T<900:
        E = 0.07-0.02*((T-900)/100)
    elif T >=900 and T<1000:
        E = 0.04-0.03*((T-1000)/100)
    elif T >=1000 and T<1100:
        E = 0.02-0.02*((T-1100)/100)
    elif T >=1100 and T<=1200:
        E = 0.02*((1200-T)/100)
    else:
        E = 0
    return (E)
```

```
def Soucinitel_E2 (T):
    if T >=20 and T<100:
        E = 1
    elif T >=100 and T<200:
        E = 0.87-0.13*((T-200)/100)
    elif T >=200 and T<300:
        E = 0.72-0.15*((T-300)/100)
    elif T >=300 and T<400:
        E = 0.56-0.16*((T-400)/100)
    elif T >=400 and T<500:
        E = 0.40-0.16*((T-500)/100)
    elif T >=500 and T<600:
        E = 0.24-0.16*((T-600)/100)
    elif T >=600 and T<700:
        E = 0.08-0.16*((T-700)/100)
    elif T >=700 and T<800:
        E = 0.06-0.02*((T-800)/100)
    elif T >=800 and T<900:
        E = 0.05-0.01*((T-900)/100)
    elif T >=900 and T<1000:
        E = 0.03-0.02*((T-1000)/100)
    elif T >=1000 and T<1100:
        E = 0.02-0.01*((T-1100)/100)
    elif T >=1100 and T<=1200:
        E = 0.02*((1200-T)/100)
    else:
        E = 0
    return (E)
```

Obr. 80 - Poměr sklonu v lineárně pružné oblasti s modulem pružnosti (Vytvořeno v prostředí Python podle [5])

Výstup (Obr. 81):



Obr. 81 - Graf poměru sklonu v lineárně pružné oblasti s modulem pružnosti v závislosti na teplotě (Vytvořeno v Prostředí python podle [5])

3.2 Modelování proužkové metody za běžné teploty

Pro modelování budu potřebovat vstupní údaje a ruční výpočet momentu únosnosti (Obr. 82):

```
b = 1000e-3 # m Širka prurezu
h = 260e-3 # m Vyska prurezu
d = 220e-3 # m Ucinna vyska prurezu
r = 5e-3 # m Polomer vyztuze
pp = 10 # - Pocet prutu
fyk = 500e6 # Pa char. mez kluzu vyztuze
fck = 30e6 # Pa char. pevnost betonu

# ##### Beton #####
fcd = fck/1.5 # Pa navrhova pevnost betonu
eps_cu = 3.5e-3 # - mezni pretvoření betonu

# ##### Ocel #####
As = pp*(r**2)*np.pi # m2 Plocha vyztuze
fyd = fyk/1.15 # Pa Navrhova mez kluzu vyztuze
Es = 200e9 # Pa Modul pružnosti oceli
Fsd = As*fyd # N Sila ve vyztuži
eyd = fyd/Es # pretvoreni na mezi kluzu oceli

# ##### Rucni vypocet
x = As * fyd / (0.8 * b * fcd)
MRd = As * fyd * (d - 0.4*x)
print("MRd1=",MRd,"x1=",x)
```

Obr. 82 - Vstupní údaje pro proužkovou metodu (Vytvořeno v prostředí Python)

Na Obr. 83 už je samotná modelace, kde si určím první polohu neutrální osy „x“, od které se bude počítat (čím větší číslo dám, tím rychlejší výpočet bude, ale musím si dát pozor, abych nezvolila takové číslo, které bude větší než skutečná vzdálenost neutrální osy od horních vláken). Dále si zvolím „dx“, což je číslo, o které se zvětšuje x do té doby, než si budou síly Fcd a Fsd rovny (samozřejmě čím menší číslo zvolím, tím přesnější výsledek dostanu, ale zase prodloužím výpočet).

```
x = 0.1
dx = 0.00001
s = 2 # 1 = bilineární, 2 = parabolicko-rektangulární
Fcd = 0
while Fcd < Fsd:
    x = x + dx
    ni = 1000 # pocet proužku
    hi = x / ni # výška proužku
    Fcd = 0
    for i in range(0,ni):
        zi = (i+0.5) * hi
        eps_ci = eps_cu * (x-zi)/x
        if s == 1:
            sigma_ci = PD_B.Bilineární(eps_ci, fcd)
        else:
            sigma_ci = PD_B.Parabolicko_rektangulární(eps_ci, fcd)
        Fci = sigma_ci*b*hi
        Fcd = Fcd + Fci
        # print(Fsd,Fcd,x)
print("Fsd=",Fsd, ",Fcd=", Fcd,"x=",x)
ni = 1000 # pocet proužku
hi = x/ni
MRd = As*fyd*(d-x)
for i in range(0,ni):
    zi = (i+0.5)*hi
    eps_ci = eps_cu*(x-zi)/x
    if s == 1:
        sigma_ci = PD_B.Bilineární(eps_ci, fcd)
    else:
        sigma_ci = PD_B.Parabolicko_rektangulární(eps_ci, fcd)
    Fci = sigma_ci * b * hi
    MRd = MRd + Fci*(x-zi)
print ("MRd2=",MRd,"x2=",x)
```

Obr. 83 - Modelování proužkové metody (Vytvořeno v prostředí Python)

Funkce „while“ mi vždy k „x“ přičte „dx“ a v tomto novém „x“ vypočítá síly a dělá to do té doby, než bude síla F_{cd} rovná nebo větší než síla F_{sd} . Ve chvíli, kdy $F_{cd} \cong F_{sd}$ vezmu x, ve kterém tato rovnost nastane a spočítám moment únosnosti. Moment se také počítá po jednotlivých proužkách – já počítám moment kolem neutrální osy, takže základní moment je od výztuže a k němu pak postupně (po proužkách) přičítám jednotlivé momenty od sil z betonu. Konečný moment únosnosti průřezu lze počítat jak z bilineárního pracovního diagramu betonu, tak z parabolicko-reaktujícího diagramu.

Konečný moment stanovený proužkovou metodou v následující kapitole porovnám s momentem únosnosti stanovený ručním (zjednodušeným) výpočtem.

3.3 Porovnání výsledků (výpočet za běžné teploty)

V této kapitole porovnám polohu neutrální osy a moment únosnosti stanovený ručním výpočtem se stejnými údaji stanovenými proužkovou metodou. Porovnání provedu pomocí mnou vytvořeného kódu popsaného výše.

Pro porovnání jsem si vybrala desku o výšce 260 mm s výztuží 10 x $\varnothing 10$ mm. V tabulce níže měním podrobnost výpočtu tím, že měním délku kroku nebo počet proužků. A srovnávám výsledky z proužkové metody s výsledky z ručního výpočtu.

Tab. 4 - Porovnání výsledků - Síly

délka kroku	počet proužků	Síla ve výztuži	Porovnáni sil					
			Ruční výpočet		Parabolicko-rekt.		Bilineární	
dx [mm]	ni [-]	Fsd [kN]	Fcd [kN]	Rozdíl sil	Fcd [kN]	Rozdíl sil	Fcd [kN]	Rozdíl sil
0,00001	1000	341,477	341,477	0,000	341,478	0,000	341,478	0,000
0,0001					341,478	-0,001	341,478	-0,001
0,001					341,490	-0,012	341,490	-0,013
0,01					341,619	-0,142	341,550	-0,073
0,1					341,619	-0,142	342,000	-0,523
1					356,191	-14,713	345,000	-3,523
1,3					361,048	-19,570	354,000	-12,523
5					420,952	-79,475	390,000	-48,523
10					501,905	-160,427	465,000	-123,523
0,0001					100	341,477	341,477	0,000
	10	341,478	-0,001	341,478	-0,001			
	1	413,439	-71,962	420,002	-78,525			

V Tab. 4 je vidět porovnání sil. Ruční výpočet má sílu v betonu stejnou jako sílu ve výztuži, to je dáno momentovou podmínkou. Výpočet přes proužkovou metodu ukazuje, že čím menší délku kroku zadám nebo čím více proužků budu mít, tak se bude síla v betonu přibližovat síle ve výztuži, až si tyto síly budou rovny.

Dále je vidět, že síly v betonu vypočítané pomocí bilineárního pracovního diagramu dávají menší hodnoty než síly počítané pomocí diagramu parabolicko-rektangulárního, což je způsobeno tím, že křivka bilineárního pracovního diagramu ze začátku roste pomaleji než křivka druhého diagramu.

Tab. 5 - Porovnání výsledků - Momenty

délka kroku dx [mm]	počet proužků ni [-]	Ruční výpočet Mrd [kN*m] x [mm]		Proužková metoda					
				Parabolicko-rektangulární			Bilineární		
				Mrd [kN*m]	Rozdíl momentů	x [mm]	Mrd [kN*m]	Rozdíl momentů	x [mm]
0,00001	1000	72,210	21,342	72,129	0,081	21,091	72,102	0,108	22,765
0,0001				72,129	0,081	21,091	72,102	0,108	22,765
0,001				72,129	0,081	21,092	72,102	0,108	22,766
0,01				72,130	0,080	21,100	72,102	0,108	22,770
0,1				72,130	0,080	21,100	72,105	0,105	22,800
1				72,189	0,021	22,000	72,120	0,090	23,000
1,3				72,212	-0,002	22,300	72,172	0,038	23,600
5				72,639	-0,429	26,000	72,443	-0,233	26,000
10				73,626	-1,416	31,000	73,348	-1,139	31,000
0,0001				100	72,210	21,342	72,129	0,081	21,091
	10	72,124	0,086	21,052			72,093	0,117	22,765
	1	72,295	-0,085	21,000			72,364	-0,154	21,000

V Tab. 5 je porovnání momentů a polohy neutrální osy. Jak je vidět, tak momenty únosnosti spočítané přesněji, tedy proužkovou, metodou jsou menší než moment spočítaný ručním (zjednodušeným) výpočtem. Což si myslím, že není zcela správně, protože když něco počítám zjednodušenou metodou, měla bych dostat nepříznivější výsledky a přesným výpočtem si tyto výsledky „vylepšit“, což se zde neděje.

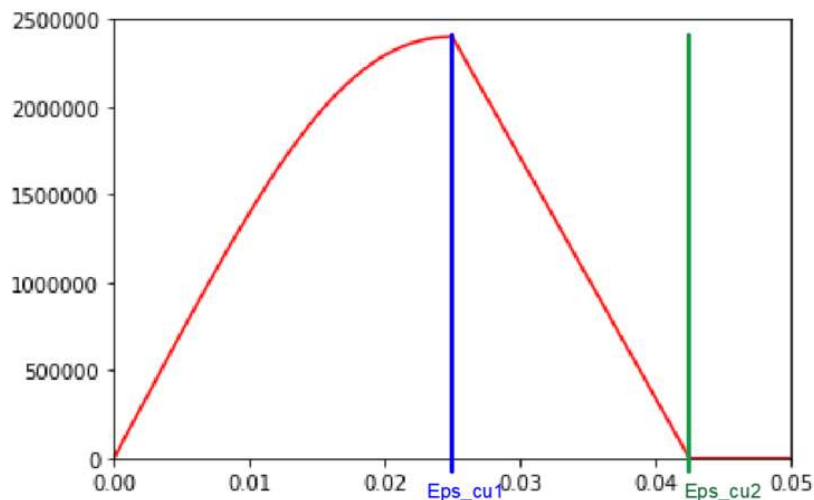
Také je vidět, že výpočet pomocí bilineárního pracovního diagramu dává větší vzdálenost neutrální osy od horních vláken a zároveň menší hodnoty celkového momentu únosnosti.

Pokud porovnáím výsledky z Tab. 4 a Tab. 5, tak zjistím, že pokud se mi rovnají síly, pak rozdíly v momentech únosnosti jsou v řádu několika desítek až stovek N·m. Pokud se mi rovnají momenty, tak pak je rozdíl v síle F_{cd} v řádu desítek kN, což značí větší nepřesnost ručního výpočtu.

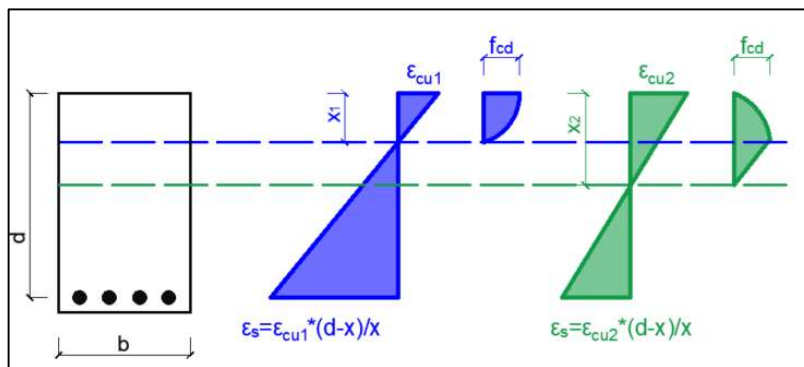
3.4 Modelování proužkové metody za požáru

Předpoklady pro použití proužkové metody za požáru:

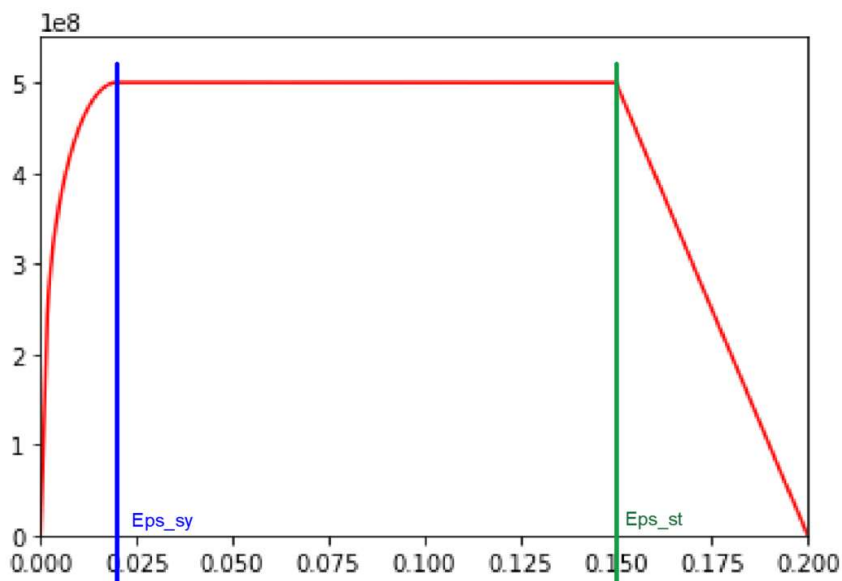
- Mezní přetvoření betonu bráno z pracovního diagramu (viz Obr. 84 a Obr. 85).
 - 1. Varianta – Mezní přetvoření bráno z maximálního napětí
 - 2. Varianta – Mezní přetvoření bráno z konce pracovního diagramu
- Přetvoření betonu je počítáno v každém proužku podle aktuální teploty.
- Přetvoření výztuže budu uvažovat pouze ve vodorovné části pracovního diagramu, tedy od $\varepsilon_{sy,\theta} = 0,02$ do $\varepsilon_{st,\theta} = 0,15$ (viz Obr. 86 a Obr. 87).
 - Tento předpoklad uvažuji proto, abych mohla počítat se stále stejnou silou ve výztuži.
 - Na Obr. 87 je vidět rozložení přetvoření v průřezu. Přetvoření betonu je dané a přetvoření oceli se pohybuje od 0,02 (modrá čára) do 0,15 (zelená čára). Tyto hranice mi určí rozsah, ve kterém budu hledat konečnou výšku tlačené oblasti. Pokud bych byla mimo tyto hranice, pak bych nemohla uvažovat konstantní sílu a tento výpočet by pak nešel použít.
- Zjednodušený výpočet momentu budu brát podle metody izoterm 500 °C.



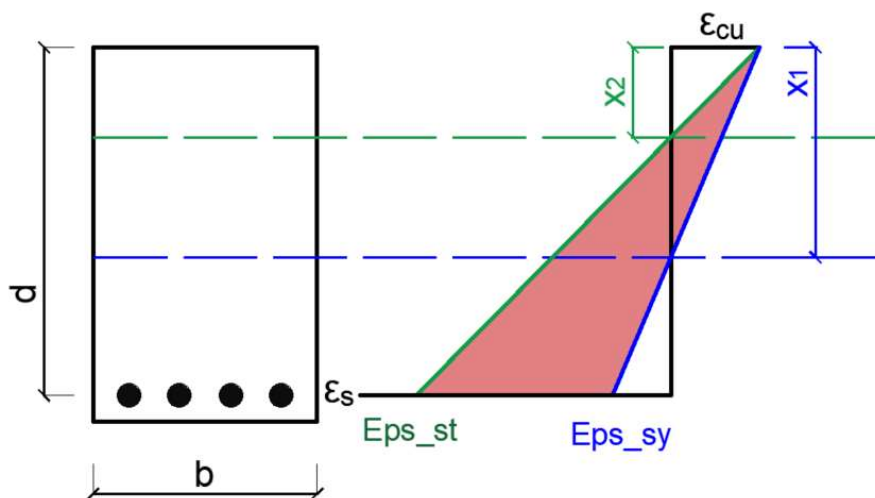
Obr. 84 - Určení mezního přetvoření v pracovním diagramu betonu (Vytvořeno v prostředí Python)



Obr. 85 - Rozložení napětí pro různé mezní přetvoření (Vytvořeno v AutoCADu)



Obr. 86 - Určení uvažovaného rozpětí přetvoření v pracovním diagramu oceli (Vytvořeno v prostředí Python)



Obr. 87 - Rozložení uvažovaného přetvoření (Vytvořeno v AutoCADu)

Pro modelování proužkové metody za požáru budu také potřebovat vstupní údaje a na rozdíl od proužkové metody za běžné teploty budu potřebovat i rozložení teploty v průřezu, které převezmu z mého kódu pro přestup tepla (Obr. 88).

```
''' Vstupní údaje '''
''' m, Pa, N '''

b = 1000e-3 # m Širka průřezu
h = 260e-3 # m Vyska průřezu
d = 220e-3 # m Učinná vyska průřezu
r = 5e-3 # m Polomer vyztuze
pp = 10 # - Pocet prutu
fyk = 500e6 # Pa char. mez kluzu vyztuze
fck = 30e6 # Pa char. pevnost betonu

##### Beton #####
fcd = fck/1.5 # Pa navrhova pevnost betonu
gamc = 1
fcd20 = fck/gamc

##### Ocel #####
As = pp*(r**2)*np.pi # m2 Plocha vyztuze
fyd = fyk/1.15 # Pa Navrhova mez kluzu vyztuze
Es = 200e9 # Pa Modul pružnosti oceli
eyd = fyd/Es # pretvoreni na mezi kluzu oceli
# print(As)

'''Zadaná teplota dle sdílení tepla'''
dx = 0.01 # m, delka vzdalenostního kroku
nx = int(h/dx) # pocer kroku
x_krok = np.linspace(0,h,nx+1) # jednotlivé kroky
T = np.array([970.4294422753027, 773.5919482656741, 615.76039
```

Obr. 88 - Vstupní údaje pro proužkovou metodu za požáru (Vytvořeno v prostředí Python)

Dále budu potřebovat, aby kód uměl ze zadaného listu teplot, spočítat teplotu v konkrétním místě. To se provede pomocí interpolace (viz Obr. 89).

```
def interpolaceT(x_krok,T,x):
    for i in range(len(x_krok)):
        if x_krok[i] > x:
            break
    dT = (T[i-1]-T[i])/(x_krok[i]-x_krok[i-1]) * (x-x_krok[i-1])
    Tx = T[i-1]-dT
    return(Tx)

T_kraj = interpolaceT(x_krok, T, 0) # teplota na okraji betonu
T_pr = interpolaceT(x_krok,T,d) # teplota prutu - vsechny pruty v 1 rovine = stejna teplota
```

Obr. 89 - Interpolace teplot (Vytvořeno v prostředí Python)


```

Varianta_eps = 1 # 1 = mezní přetvoření v místě max. napětí, 2 = mezní přetvoření na konci prac.diagramu
if Varianta_eps == 1:
    eps_cu = PD_Bpozar.ec1_temper(T_kraj)
    print("eps_c=",eps_cu)
if Varianta_eps == 2:
    eps_cu = PD_Bpozar.ecul_temper(T_kraj)
    print("eps_cu=",eps_cu)

### pevnost jednotlivých prutu #####
gams = 1

print(T_pr)
k = 3 # součinitel ks
if k == 1: #vyztuz valcovana za tepla
    ks = PD_Opozar.Soucinitel_k1(T_pr)
elif k == 2: #vyztuz valcovana za studena
    ks = PD_Opozar.Soucinitel_k2(T_pr)
elif k == 3: #tlakova a tahova vyztuz
    ks = PD_Opozar.Soucinitel_k3(T_pr)
fydfi = ks*(fyk/gams)

# # ##### Rucni vypocet #####
ny = 0.7
x = (As * fydfi) / (0.8 * b * fcd20 * ny)
MRd = As * fydfi * (d - 0.4*x)
print("MRd1=",np.ceil(1000*(MRd/1000))/1000,"kNm",", x1=",np.ceil(1000*(x*1000))/1000,"mm")

```

Obr. 90 - Proužková metoda za požáru - přetvoření, součinitel ks , moment únosnosti (Vytvořeno v prostředí Python)

Na Obr. 90 lze vidět zvolenou variantu, výběr součinitele ks podle typu výztuže a určení momentu únosnosti pomocí zjednodušeného výpočtu.

Na Obr. 91 je vidět zbytek kódu, kde probíhá výpočet momentu únosnosti za požární situace pomocí proužkové metody.

```

eps_s1 = 0.02
eps_s2 = 0.15
x0 = (d*eps_cu)/(eps_cu+eps_s2)
x_end = (d*eps_cu)/(eps_cu+eps_s1)
print("x0=",x0,"mm, x_end = ",x_end,"mm")
kc = 1 # součinitel kc (1=křemičité kamenivo, 2=vápencové kamenivo)
x = x0
dx = 0.1
Fcd = 0
ni = 1000 # pocet prouzku
while Fcd < Fsd:
    x = x + dx
    if x > x_end:
        print("chyba: x přesáhlo hodnotu x_end")
        break
    hi = x / ni # výška prouzku
    Fcd = 0
    for i in range(0,ni):
        zi = (i+0.5) * hi
        Ti = interpolateT(x_krok,T,zi)
        eps_ci = eps_cu * (x-zi)/x
        sigma_ci = PD_Bpozar.Pozar_diagram(eps_ci, Ti, fck, kc)
        Fci = sigma_ci*b*hi
        Fcd = Fcd + Fci
        # print(Fsd,Fcd,x,Ti)
print("Fsd=",np.ceil(1000*(Fsd/1000))/1000,"kN", ",Fcd=", np.ceil(1000*(Fcd/1000))/1000,"kN",\
      ",x=",np.ceil(1000*(x*1000))/1000,"mm")
hi = x/ni
MRd = As*fydfi*(d-x)
for i in range(0,ni):
    zi = (i+0.5)*hi
    Ti = interpolateT(x_krok,T,zi)
    eps_ci = eps_cu*(x-zi)/x
    sigma_ci = PD_Bpozar.Pozar_diagram(eps_ci, Ti, fck, kc)
    Fci = sigma_ci * b * hi
    MRd = MRd + Fci*(x-zi)
print ("MRd2=",np.ceil(1000*(MRd/1000))/1000,"kNm",",x2=",np.ceil(1000*(x*1000))/1000,"mm")

```

Obr. 91 - Proužková metoda za požáru - výpočet (Vytvořeno v prostředí Python)

Výstupem z toho kódu jsou momenty únosnosti za požární situace, spočítané zjednodušeným výpočtem a proužkovou metodou. Důležitým údajem je i poloha neutrální osy „x“ (Obr. 92).

```

eps_c= 0.025
Teplota prutu = 22.08 °C
MRd1= 82.723 kNm ,x1= 23.375 mm
Fsd= 341.478 kN ,Fcd= 347.736 kN ,x= 28.001 mm
MRd2= 76.935 kNm ,x2= 28.001 mm

```

Obr. 92 - Výstup z proužkové metody za požáru (Vytvořeno v prostředí Python)

3.5 Porovnání výsledků (výpočet za požární situace)

V této části porovnám polohu neutrální osy a moment únosnosti stanovený ručním výpočtem se stejnými údaji stanovenými proužkovou metodou, pro dva typy mezního přetvoření. Porovnání provedu pomocí mnou vytvořeného kódu popsaného výše.

Pro porovnání jsem si vybrala desku o výšce 260 mm s výztuží 10 x Ø10 mm, která bude vystavena požáru 90 minut podle normové teplotní křivky, nejprve z jedné strany (dále od výztuže), následně z druhé strany (blíže k výztuži) a nakonec z obou stran.

V prvním případě půjde o mezní přetvoření v místě maximálního napětí. Ve druhém případě půjde o mezní přetvoření na konci pracovního diagramu

V tabulce níže měním podrobnost výpočtu tím, že měním délku kroku nebo počet proužků. A srovnávám výsledky z proužkové metody s výsledky z ručního výpočtu (metoda IZO500).

- Varianta 1: Požár zespod (u výztuže)
 - o Teplota na dolním okraji: 970,4 °C
 - o Teplota na horním okraji: 20,7 °C
 - o Teplota výztuže: 391,3 °C
- Varianta 2: Požár ze shora (dál od výztuže)
 - o Teplota na dolním okraji: 20,7 °C
 - o Teplota na horním okraji: 970,4 °C
 - o Teplota výztuže: 22,1 °C
- Varianta 3: Požár z obou stran
 - o Teplota na dolním okraji: 970,4 °C
 - o Teplota na horním okraji: 970,4 °C
 - o Teplota výztuže: 392,7 °C

Tab. 6 - Porovnání momentů únosnosti za požáru pomocí proužkové metody

délka kroku [m]		počet proužků		Ruční výpočet		Proužková metoda						Síla ve výztuži (ruční výpočet)		E(max. napětí)		E(konec diagramu)										
dx	[m]	ni	[-]	Mrd	[kN*m]	Mrd	[kN*m]	Rozdíl momentů	x	[mm]	Mrd	[kN*m]	Rozdíl momentů	x	[mm]	Fsd	[kN]	Fcd	[kN]	Rozdíl sil	Fsd	[kN]	Fcd	[kN]	Rozdíl sil	
0,00001																										
0,0001																										
0,001		1000		59,387		59,762	-0,375		14,626		57,823	1,564		25,918		278,339				-0,015		401,907			-123,583	
0,01						61,443	-2,056		23,626		57,843	1,544		26,908		278,324						417,258			-138,934	
0,1																						449,465			-171,141	
0,0001		100				59,762	-0,375		14,626		57,824	1,563		26,008		278,341						403,348			-125,024	
0,0001		10		59,387		59,760	-0,373				57,830	1,557				278,626						406,533			-128,209	
		1				59,402	-0,015		13,226		59,797	-0,410				279,948						446,297			-167,973	

délka kroku [m]		počet proužků		Ruční výpočet		Proužková metoda						Síla ve výztuži (ruční výpočet)		E(max. napětí)		E(konec diagramu)									
dx	[m]	ni	[-]	Mrd	[kN*m]	Mrd	[kN*m]	Rozdíl momentů	x	[mm]	Mrd	[kN*m]	Rozdíl momentů	x	[mm]	Fsd	[kN]	Fcd	[kN]	Rozdíl sil	Fsd	[kN]	Fcd	[kN]	Rozdíl sil
0,00001																									
0,0001																									
0,001		1000		71,248		76,091	-4,843		45,259		75,833	-4,585		50,135		392,740				-0,041		538,850			-146,151
0,01						76,104	-4,856		45,329		75,827	-4,579		50,225		392,699						539,942			-147,243
0,1						77,833	-6,585		51,429		75,223	-3,975		60,125								550,721			-158,022
0,0001		100				76,104	-4,856		45,329		75,827	-4,579		50,225		392,699						539,985			-147,286
0,0001		10		71,248		76,103	-4,855		45,129		75,814	-4,566				393,160						544,278			-151,579
		1				78,200	-6,952		41,829		83,565	-12,317				393,582						672,731			-280,032

délka kroku [m]		počet proužků		Ruční výpočet		Proužková metoda						Síla ve výztuži (ruční výpočet)		E(max. napětí)		E(konec diagramu)									
dx	[m]	ni	[-]	Mrd	[kN*m]	Mrd	[kN*m]	Rozdíl momentů	x	[mm]	Mrd	[kN*m]	Rozdíl momentů	x	[mm]	Fsd	[kN]	Fcd	[kN]	Rozdíl sil	Fsd	[kN]	Fcd	[kN]	Rozdíl sil
0,00001																									
0,0001																									
0,001		1000		51,133		54,833	-3,700		39,879		56,309	-5,176		50,136		277,881				-0,128		538,173			-260,420
0,01						54,840	-3,707		39,929		56,314	-5,181		50,226		277,753						539,270			-261,517
0,1						54,711	-3,578		40,429		56,362	-5,229		51,126								550,039			-272,286
						55,075	-3,942		41,429		56,856	-5,723		60,126								650,279			-372,526
0,0001		100				54,840	-3,707		39,929		56,314	-5,181		50,226		277,753						539,327			-261,574
0,0001		10		51,133		54,848	-3,715		39,829		56,305	-5,172				279,199						543,779			-266,026
		1				56,213	-5,080		35,429		64,109	-12,976				279,279						675,091			-397,338

Požár ze spoda

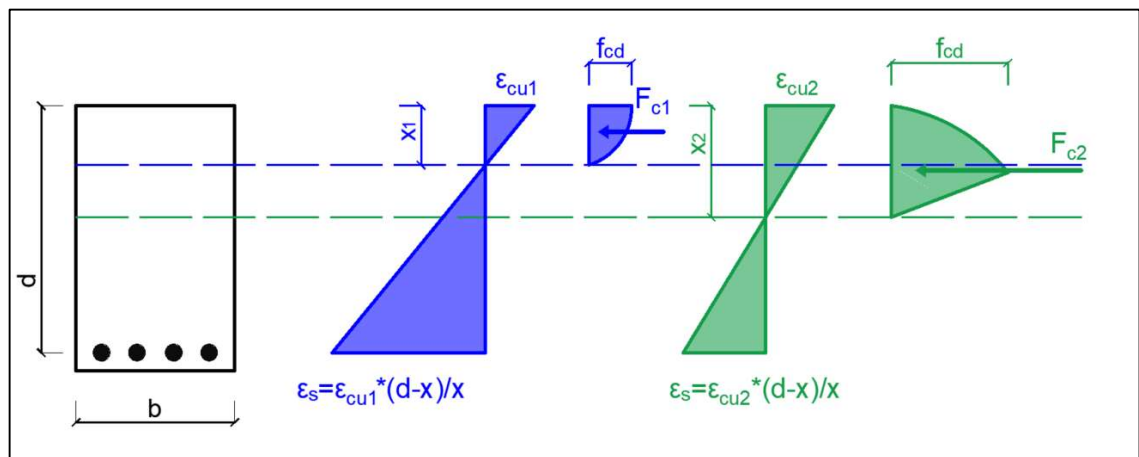
Požár ze shora

Požár z obou stran

V Tab. 6 jsem provedla porovnání momentů únosnosti, polohy neutrální osy a sil. Vždy jsem porovnávala hodnoty stanovené zjednodušeným výpočtem s hodnotami vypočítanými z proužkové metody. V tomto porovnání jsem přišla na několik zvláštností.

První zarážející hodnoty jsou hodnoty sil, počítaných z mezního přetvoření na konci diagramu. Tyto síly v betonu vychází až 2x větší, než je síla ve výztuži, přitom momenty únosnosti vychází dosti podobně. Proč tomu tak je vysvětlím na následujícím obrázku (Obr. 93).

Jak je vidět na obrázku, pokud počítám s mezním přetvořením od maximálního napětí (modrá barva), je výška tlačené oblasti menší a je menší i celkové napětí, proto bude menší i síla, která se bude blížit síle ve výztuži, a to proto, že tvar napětí v tomto případě je podobný tvaru napětí za běžné teploty. Pokud počítám s mezním přetvořením z konce pracovního diagramu (zelená barva), pak budu mít větší výšku tlačené oblasti, větší napětí a tím i větší celkovou sílu. V tomto případě se mi síla ve výztuži a síla v betonu opravdu rovnat nemůže.



Obr. 93 - Rozložení napětí v průřezu (Vytvořeno v AutoCADu)

Další zarážející hodnotou jsou hodnoty momentu únosnosti, které se zvyšující se přesností klesají, což by znamenalo, že čím detailnější výpočet provedu, tím bude únosnost prvku menší. Toto tvrzení nedává smysl a vyvrací veškeré logické myšlení. Proto si myslím, že tato proužková metoda není vhodná pro posouzení momentů únosnosti za požární situace.

Tato proužková metoda za požáru dává také větší rozdíly momentů než proužková metoda za běžné teploty, a to hlavně v případech, kdy hoří ze shora, tedy u tlačené části betonu. Celkově to jsou pořád malé rozdíly, v řádu pár jednotek kNm, což znamená že se ve výpočtu nevyskytuje žádná zásadní chyba.

Kapitola 4

Praktická část

Praktická část je rozdělena na 3 bloky, přičemž každý je věnován popisu jiné problematiky.

1. První blok se věnuje sestrojení kódu pro 1D modelování přestupu tepla skrz konstrukce.
2. Druhý blok se věnuje sestrojení kódu pro výpočet momentu únosnosti pomocí proužkové metody.
3. Třetí blok se věnuje sestrojení kódu pro posouzení požární odolnosti podle metody izotermy 500 °C a zónové metody.

4 Praktická část – 3. blok

Ve 3. bloku praktické části se budu věnovat sestrojení kódu pro posouzení požární odolnosti podle metody izotermy 500 °C a zónové metody.

Uvedu zde kódy z programovacího jazyka pro jednotlivé součinitele potřebné k výpočtům a také pro výpočty požární odolnosti podle uvedených metod.

Na závěr provedu porovnání výsledků obou zmíněných metod.

4.1 Izoterma 500 °C

Zde uvedu kód pro modelování izotermy 500 °C. Pro vytvoření kódu budu potřebovat vstupní údaje, rozložení teploty po průřezu, které převezmu z mého kódu pro přestup tepla a součinitel k_s pro redukci charakteristické pevnosti oceli, který je popsán již ve 2. praktické části.

V metodě izotermy se redukuje pouze ocel a u betonu se předpokládá, že pokud nepřesáhne 500°C, pak má stejnou únosnost jako při 20 stupňů, proto mě bude zajímat vzdálenost od kraje průřezu, ve které je 500 °C a teplota v prutu.

Na obrázcích níže je znázorněn postup modelování kódu. Nejprve jsou uvedeny vstupní údaje, kterými jsou rozměry desky/stěny a materiálové vlastnosti (Obr. 94). Na Obr. 95 je zadáno rozložení teploty v konstrukci. Toto rozložení teploty jsem zkopírovala ze svého kódu, popsaného výše (v první praktické části) a pomocí připravené interpolace dokážu získat teplotu ve kterémkoli bodě – např. výztuži.

```
##### Rozměry a materiály #####
## deska/stena ##
b = 1000e-3      # m sirka (1 m bezny)
h = 260e-3      # m tloustka
d = 220e-3      # m ucinna vyska
## beton ##
fck = 30e6      # Pa char. pevnost betonu (např. 25 = C25/30)
## ocel ##
fyk = 500e6     # Pa char. mez kluzu vyztuze (B500B)
pp = 10        # pocet prutu
r = 5e-3       # m polomer prutu
As = pp*(r**2)*np.pi # mm2 plocha vyztuze
# As = 2324
```

Obr. 94 - Vstupní údaje - rozměry a materiály (Vytvořeno v prostředí Python)

```
##### pozar #####
dx = 0.01      # m, delka vzdalenostniho kroku
nx = int(h/dx) # pocer kroku
x_krok = np.linspace(0,h,nx+1) # jednotlivé kroky
T = np.array([970.3837544896085, 773.3063149150731, 615.2033270076344, 490.24570])
def interpolaceT(x_krok,T,x):
    for i in range(len(x_krok)):
        if x_krok[i] > x:
            break
    dT = (T[i-1]-T[i])/(x_krok[i]-x_krok[i-1]) * (x-x_krok[i-1])
    Tx = T[i-1]-dT
    return(Tx)
```

Obr. 95 - Zadání požáru - teploty převzaty z mého kódu pro přestup tepla (Vytvořeno v prostředí Python)

Na Obr. 96 je znázorněna volba pro zadání požáru. Řádek „pozar =“ udává z jaké strany je požár, vtaženo k umístění výztuže. Dále se musí zadat „zadani_pozaru“, které udává, jak je požár zadán. Pomocí interpolace zde zjistím vzdálenost od kraje, kde bude 500 °C a také teplotu ve výztuži.

```

##### Pozar #####
bfi = b

pozar = 2 # 1= pozar u vyztuze, 2= pozar dál od vyztuze, 3= pozar z obou stran
zadani_pozaru = 2 # 1= sestupně [900,800,700,...], 2= vzestupně [20,100,200,...]

if zadani_pozaru == 1:
    def interpolateX1(T, x_krok, t):
        for i in range(len(T)):
            if T[i] < t:
                break
            dx = x_krok[i]-(t-T[i])*(x_krok[i]-x_krok[i-1]) / (T[i-1]-T[i])
        return(dx)
    x_500 = interpolateX1(T, x_krok, 500)
    a500 = x_500
elif zadani_pozaru == 2:
    def interpolateX2(T, x_krok, t):
        for i in range(len(T)):
            if T[i] > t:
                break
            dx = x_krok[i]-(t-T[i])*(x_krok[i]-x_krok[i-1]) / (T[i-1]-T[i])
        return(dx)
    x_500 = interpolateX2(T, x_krok, 500)
    a500 = h - x_500
T2=interpolateT(x_krok, T,x_500)
print("Teplota ",np.ceil(100*T2)/100,"°C, ve vzdálenosti",np.ceil(1000*x_500*1000)/1000,"mm (bráno zleva)")

if zadani_pozaru ==1 and pozar ==1:
    T1=interpolateT(x_krok, T,h-d)
elif zadani_pozaru ==1 and pozar ==2:
    T1=interpolateT(x_krok, T,d)
elif zadani_pozaru ==2 and pozar ==1:
    T1=interpolateT(x_krok, T,d)
elif zadani_pozaru ==2 and pozar ==2:
    T1=interpolateT(x_krok, T,h-d)
print("Teplota ve výztuži=",np.ceil(100*T1)/100,"°C")

if pozar == 1: # Pozar z 1 strany - tazana strana [u vyztuze]
    dfi = d
    print("dfi= d=",np.ceil(1000*dfi*1000)/1000,"mm")
elif pozar == 2: # Pozar z 1 strany - tlacena strana [dal od vyztuze]
    dfi = d - a500
    print("dfi=",np.ceil(1000*dfi*1000)/1000,"mm")
elif pozar == 3: # Pozar ze 2 stran
    dfi = d - a500
    hfi = h - 2*a500
    print("dfi=",np.ceil(1000*dfi*1000)/1000,"mm")
hfi = h - a500

```

Obr. 96 - Charakteristika požáru (Vytvořeno v prostředí Python)

Na Obr. 97 počítám pevnost betonu a pevnost prutů. Vzhledem k tomu, že tento výpočet provádím pro desku/stěnu, uvažuji s výztuží pouze v jedné rovině, proto mě bude zajímat pouze jedna teplota ve výztuži. Dále si zde vypočítám výšku tlacené oblasti.

```

#### pevnost betonu ####
gamc = 1
fcd20 = fck/gamc

#### pevnost jednotlivych prutu ####
gams = 1
Tpr = T1 # teplota prutu - vsechny pruty v 1 rovine = stejna teplota
k = 3 # soucinitel ks
def Soucinitel_k1 (T):
def Soucinitel_k2 (T):
def Soucinitel_k3 (T):
if k == 1: #vyztuz valcovana za tepla
    ks = Soucinitel_k1(Tpr)
elif k == 2: #vyztuz valcovana za studena
    ks = Soucinitel_k2(Tpr)
elif k == 3: #tlakova a tahova vyztuz
    ks = Soucinitel_k3(Tpr)

fydfi = ks*(fyk/gams)

##### tlacena oblast #####
lam = 0.8
ny = 0.7

xfi = (As*fydfi)/(bfi*lam*ny*fcd20) # m
print("xfi =", np.ceil(1000*(xfi*1000))/1000, "mm")

```

Obr. 97 - Součinitel ks (Vytvořeno v prostředí Python podle[5])

Na Obr. 98 je vidět výpočet síly ve výztuži a momentu únosnosti. Dále je zde údaj o zatížení při požární situaci a následné porovnání, zda průřez na takovéto zatížení vyhoví.

```
##### vysledna sila a moment ve vyztuži #####
Fsdfi = As*fydfi # N
zfi = dfi - 0.5*lam*xfi # m

MRdfi = (Fsdfi*zfi) # Nm
MRdfi_kNm = MRdfi/1000

##### zatizeni pri pozaru #####
Ed = 13.911e3 # Nm navrhova hodnota zatizeni za bezne teploty
nyfi = 0.7 # redukcní soucinitel
Edfi = nyfi*Ed
Edfi_kNm = Edfi/1000

##### Posouzení #####

if (MRdfi > Edfi):
    print("Konstrukce vyhoví na zatížení při požární situaci.")
    print("Návrhová hodnota únosnosti prvku =", np.ceil(1000*MRdfi_kNm)/1000, "kNm")
    print("Návrhová hodnota zatížení při požáru =", np.ceil(1000*Edfi_kNm)/1000, "kNm")
    print("\U0001F603",np.ceil(100*MRdfi_kNm)/100,">",np.ceil(100*Edfi_kNm)/100,"\U0001F603")
else:
    print("Konstrukce NEVYHOVÍ na zatížení při požární situaci.")
    print("Návrhová hodnota únosnosti prvku =", np.ceil(1000*MRdfi_kNm)/1000, "kNm")
    print("Návrhová hodnota zatížení při požáru =", np.ceil(1000*Edfi_kNm)/1000, "kNm")
    print("\U0001F922",np.ceil(100*MRdfi_kNm)/100,"<",np.ceil(100*Edfi_kNm)/100,"\U0001F922")
```

Obr. 98 - Výpočet síly a momentu (Vytvořeno v prostředí Python podle [5])

Na Obr. 99 je vidět výstup z mého kódu, kde jsem nechala vypsat vzdálenost od okraje, kde se nachází 500°C, dále jaká je teplota ve výztuži, účinnou výšku za požáru, výšku tlačené oblasti za požární situace a následně porovnání momentů únosnosti.

```
Teplota 500.0 °C, ve vzdálenosti 0.03 mm
Teplota ve výztuži= 22.08 °C
dfi= 190.781 mm
xfi = 23.375 mm
Konstrukce vyhoví na zatížení při požární situaci.
Návrhová hodnota únosnosti prvku = 71.248 kNm
Návrhová hodnota zatížení při požáru = 9.738 kNm
😄 71.25 > 9.74 😄
```

Obr. 99 - Výsledky z mého kódu (Vytvořeno v prostředí Python)

Tímto kódem se dá ověřit požární odolnost železobetonové desky nebo stěny. V navazující kapitole popíšu zónovou metodu a na závěr této části srovnám výsledky obou těchto metod.

4.2 Zónová metoda

Zde uvedu kód pro modelování zónové metody. Pro vytvoření kódu budu potřebovat vstupní údaje, rozložení teploty po průřezu, které převezmu z mého kódu pro přestup tepla a součinitele k_c a k_s pro redukci charakteristické pevnosti betonu a oceli, jež jsou popsány již ve 2. praktické části.

V zónové metodě se redukuje jak ocel, tak i beton. Zde mě budou zajímat teploty jak v prutu, tak i v jednotlivých zónách.

Na obrázcích níže je znázorněn postup modelování kódu. Nejprve jsou uvedeny vstupní údaje, kterými jsou rozměry desky/stěny a materiálové vlastnosti (Obr. 100). Na Obr. 101 se volí, zda je požár z jedné strany nebo z obou stran a také počet zón.

```
##### Rozměry a materiály #####
## deska/stena ##
h = 260e-3      # m výška průřezu
b = 1000e-3     # m šířka průřezu
## materiály ##
fck = 30e6      # Pa charakteristická pevnost betonu
fyk = 500e6     # Pa charakteristická mez kluzu výztuže
d = 220e-3     # m účinná výška
pp = 10        # - počet prutu
r = 5e-3       # m polomer prutu

As = pp*(r**2)*np.pi # m2 plocha vyztuže
# As = 2324
```

Obr. 100 - Vstupní údaje - rozměry a materiály (Vytvořeno v prostředí Python)

```
##### Pozar #####
pozar_str = 1

if pozar_str == 1:      # Pozar z 1 strany
    w = h
elif pozar_str == 2:   # Pozar ze 2 stran
    w = (1/2)*h
n = 4                  # Pocet zon (min.3)
zony = np.arange(1,n+1,1)
```

Obr. 101 - Určení požáru a počtu zón (Vytvořeno v prostředí Python)

Na následujícím obrázku (Obr. 102) je zadáno rozložení teploty v konstrukci. Toto rozložení teploty jsem zkopírovala ze svého kódu, popsaného výše (v první praktické části) a pomocí připravené interpolace dokážu získat teplotu ve kterémkoli bodě – např. výztuži či jednotlivých zónách.

```
##### Zadaná teplota dle sdílení tepla #####
dx = 0.01 # m, délka vzdalenostního kroku
nx = int(h/dx) # počet kroku
x_krok = np.linspace(0,h,nx+1) # jednotlivé kroky
# T = np.array([20.703809314742163, 20.807078543872574, 21.046656
T = np.array([970.3837544896085, 773.3063149150731, 615.203327007
def interpolaceT(x_krok,T,x):
    for i in range(len(x_krok)):
        if x_krok[i] > x:
            break
    dT = (T[i-1]-T[i])/(x_krok[i]-x_krok[i-1]) * (x-x_krok[i-1])
    Tx = T[i-1]-dT
    return(Tx)
```

Obr. 102 - Rozložení teploty v průřezu - převzato z mého kódu pro přestup tepla v konstrukci (Vytvořeno v prostředí Python)

Na Obr. 103 počítám jednotlivé teploty ve středu každé zóny, součinitele redukce a také tloušťku vyloučené vrstvy.

```
##### rozmery zón #####
sirka_zony = w/n
stred_zony = sirka_zony/2

##### Teploty ve středu jednotlivých vrstev #####
Ti = []
Ti.append(interpolaceT(x_krok, T, stred_zony)) # Teplota v 1. zóně
for i in np.arange(1,n,1):
    zona_i = stred_zony + i*sirka_zony
    Ti.append(interpolaceT(x_krok, T, zona_i)) # Teplota v prostředních zónách
T_M =interpolaceT(x_krok, T, w) # Teplota v bodě M
#####
kcc = 1
def Soucinitel_k1 (T): # Beton s křemičitým kamenivem
def Soucinitel_k2 (T): # Beton s vápencovým kamenivem
#####
kci = []
for teplota in Ti:
    if kcc == 1: #obycejny beton s kremicitym kamenivem
        kci.append(Soucinitel_k1(teplota))
    if kcc == 2: #obycejny beton s vapencovym kamenivem
        kci.append(Soucinitel_k2(teplota))
kc_M = Soucinitel_k2(T_M)
#####
kc_m = ((1-0.2/n)/n)*sum(kci) # - střední součinitel redukce
#####
a_z = w*(1-(kc_m/kc_M)) # m tloušťka vyloučené vrstvy pro nosníky,
# desky a smykové stěny bez účinků 2.řádu
```

Obr. 103 - Určení teplot v jednotlivých zónách (Vytvořeno v prostředí Python podle [5])

Na následujícím obrázku (Obr. 104) je znázorněna volba pro zadání požáru. Řádek „pozar“ udává z jaké strany je požár (vtaženo k umístění výztuže). Dále se musí zadat „zadani_pozaru“, které udává, jak je požár zadáný.

```

# #### pevnost betonu ####
gamc = 1
fcdfi = kc_M*fck/gamc
#### pevnost jednotlivych prutu ####
gams = 1
pozar = 1 # 1= pozar u vyztuze, 2= pozar dál od vyztuze, 3= pozar z obou stran
zadani_pozaru = 1 # 1= sestupně [900,800,700,...], 2= vzestupně [20,100,200,...]
if zadani_pozaru ==1 and pozar ==1:
    T1=interpolaceT(x_krok, T,h-d)
elif zadani_pozaru ==1 and pozar ==2:
    T1=interpolaceT(x_krok, T,d)
elif zadani_pozaru ==2 and pozar ==1:
    T1=interpolaceT(x_krok, T,d)
elif zadani_pozaru ==2 and pozar ==2:
    T1=interpolaceT(x_krok, T,h-d)
Tpr = T1 # teplota prutu - vsechny pruty v 1 rovine = stejna teplota
print("Teplota ve výztuži =",np.ceil(100*Tpr)/100,"°C")
def Soucinitel_k1 (T):
def Soucinitel_k2 (T):
def Soucinitel_k3 (T):
k = 3 # soucinitel ks
if k == 1: #vyztuz valcovana za tepla
    ks = Soucinitel_k1(Tpr)
elif k == 2: #vyztuz valcovana za studena
    ks =Soucinitel_k2(Tpr)
elif k == 3: #tlakova a tahova vyztuz
    ks =Soucinitel_k3(Tpr)
#####
fydfi = ks*(fyk/gams)

```

Obr. 104 - Určení pevnosti prutů (Vytvořeno v prostředí Python podle [5])

Na Obr. 105 je proveden výpočet tlačené oblasti, síly ve výztuži a následně i momentu únosnosti. V posledních řádcích je opět porovnání momentu únosnosti s momentem od zatížení.

```

##### tlacena oblast #####
lam = 0.8
ny = 0.7

xfi = (As*fydfi)/(b*lam*ny*fcdfi) # m
print("xfi =", np.ceil(1000*(xfi*1000))/1000, "mm")

##### vysledna sila a moment ve vyztuži #####
Fsdfi = As*fydfi # N
zfi = d - 0.5*lam*xfi # m
print("Síla ve výztuži = ",np.ceil(1000*Fsdfi/1000)/1000,"kN, na rameni", np.ceil(100*zfi*1000)/100,"mm")
MRdfi = (Fsdfi*zfi) # Nm
MRdfi_kNm = MRdfi/1000 # kNm

##### zatizeni pri pozaru #####

Ed = 13.911e3 # Nm navrhova hodnota zatizeni za bezne teploty
nyfi = 0.7 # redukčni soucinitel
Edfi = nyfi*Ed # Nm navrhova hodnota zatizeni za pozaru
Edfi_kNm = Edfi/1000 # kNm

##### Posouzení #####

if (MRdfi > Edfi):
    print("Konstrukce vyhoví na zatížení při požární situaci.")
    print("Návrhová hodnota únosnosti prvku =", np.ceil(1000*MRdfi_kNm)/1000, "kNm")
    print("Návrhová hodnota zatížení při požáru =", np.ceil(1000*Edfi_kNm)/1000, "kNm")
    print("\U0001F603",np.ceil(100*MRdfi_kNm)/100,">",np.ceil(100*Edfi_kNm)/100,"\U0001F603")
else:
    print("Konstrukce NEVYHOVÍ na zatižení při požární situaci.")
    print("Návrhová hodnota únosnosti prvku =", np.ceil(1000*MRdfi_kNm)/1000, "kNm")
    print("Návrhová hodnota zatížení při požáru =", np.ceil(1000*Edfi_kNm)/1000, "kNm")
    print("\U0001F922",np.ceil(100*MRdfi_kNm)/100,"<",np.ceil(100*Edfi_kNm)/100,"\U0001F922")

```

Obr. 105 - Výpočet tlačení oblasti, síly ve výztuži a momentu únosnosti (Vytvořeno v prostředí Python podle [5])

Na Obr. 106 je vidět výstup z mého kódu, kde jsem nechala vypsat teplotu ve výztuži, sílu ve výztuži, výšku tlačené oblasti za požární situace a následně porovnání momentů únosnosti.

```

Teplota ve výztuži = 391.26 °C
xfi = 16.567 mm
Síla ve výztuži = 278.324 kN, na rameni 213.38 mm
Konstrukce vyhoví na zatížení při požární situaci.
Návrhová hodnota únosnosti prvku = 59.387 kNm
Návrhová hodnota zatížení při požáru = 9.738 kNm
☺ 59.39 > 9.74 ☹

```

Obr. 106 - Výsledky z mého kódu (Vytvořeno v prostředí Python)

Tímto kódem se také dá ověřit požární odolnost železobetonové desky nebo stěny, stejně tak jako u metody 500 °C. V navazující kapitole srovnám výsledky obou těchto metod.

4.2.1 Porovnání podrobnosti výpočtu

Zde uvedu názorný příklad toho, jak se budou měnit výsledky, pokud budu mít průřez rozdělen na více zón. Pro srovnání jsem použila požár z jedné strany (dál od výztuže), v trvání 90 minut, podle normové teplotní křivky.

V tabulce níže je srovnání momentů únosnosti a tloušťky vyloučené vrstvy v závislosti na počtu zón.

Tab. 7 - Porovnání podrobnosti výpočtu zónové metody

počet zón [-]	šířka zóny [mm]	a_z [mm]	$M_{Rd,fi}$ [kNm]
3	86,67	34,72	69,09
4	65,00	34,74	69,08
6	43,33	34,43	69,20
8	32,50	33,76	69,46
10	26,00	32,60	69,92
15	17,33	31,84	70,22
20	13,00	30,78	70,63
30	8,67	29,95	70,96
50	5,20	29,41	71,17
100	2,60	28,95	71,35
200	1,30	28,72	71,44
260	1,00	28,67	71,47
500	0,52	28,58	71,50
1000	0,26	28,53	71,52

Tímto srovnáním jsem dokázala, že čím více zón budeme počítat, tím vyšší únosnost dostaneme. Jedná se ale pouze o pár jednotek kNm, takže se dá říci, že normou dané min. 3 zóny dávají dostačující výsledek. Také jsem zjistila, že mému kódu netrvá ani vteřinu spočítat moment únosnosti při 1000 zónách, z čehož vyplývá, že při použití programu se dá provádět dosti podrobný výpočet, který je rychlý a přesný.

4.3 Porovnání výstupů

Zde porovnám výsledky z výše popsaných metod. Budu mít stejné vstupní údaje, kdy rozměry a materiál desky zůstane stejný a budou se měnit parametry požáru (délka trvání požáru a strana, ze které je deska požáru vystavena). V zónové metodě budu uvažovat přesnější výpočet pro 1000 proužků.

Tab. 8 - Porovnání výstupů ze zónové metody a z metody izotermy

Požár	Maximální teplota na kraji průřezu [°C]	Teplota ve výztuži [°C]	Metoda izotermy 500°C		Zónová metoda		Rozdíly momentů [kN*m]
			Mrd [kN*m]	Vyloučená vrstva a500 [mm]	Mrd [kN*m]	Vyloučená vrstva az [mm]	
30 min	Z jedné strany (dál od výztuže)	20,0	78,791	10,01	78,406	10,79	0,385
	Z jedné strany (požár u výztuže)	150,1	78,753		78,753		0,000
	Požár z obou stran	150,1	75,018	20,02	74,726	21,59	0,292
60 min	Z jedné strany (dál od výztuže)	20,2	74,488	20,97	74,492	20,96	-0,004
	Z jedné strany (požár u výztuže)	294,4	67,216		67,216		0,000
	Požár z obou stran	294,6	60,569	41,94	60,631	41,92	-0,062
90 min	Z jedné strany (dál od výztuže)	22,1	71,248	29,22	71,465	28,67	-0,217
	Z jedné strany (požár u výztuže)	391,3	59,387		59,387		0,000
	Požár z obou stran	392,7	51,133	58,44	51,232	57,34	-0,099
120 min	Z jedné strany (dál od výztuže)	26,67	68,427	36,81	68,728	35,72	-0,301
	Z jedné strany (požár u výztuže)	463,81	51,912		51,912		0,000
	Požár z obou stran	486,34	42,595	73,62	42,843	71,44	-0,248
180 min	Z jedné strany (dál od výztuže)	43,02	63,772	48,26	64,157	47,28	-0,385
	Z jedné strany (požár u výztuže)	570,15	34,402		34,402		0,000
	Požár z obou stran	586,1	23,920	96,51	24,451	94,55	-0,531

Pro porovnání výsledků jsem si vybrala desku tl. 260 mm, s výztuží 10 x Ø 10 mm, z betonu C 30/37 a oceli B500B. Jak je vidět z tabulky výše, rozdíly momentů počítaných zónovou metodou nebo metodou izotermie se v případě požáru u výztuže shodují, což je dáno tím, že výztuž redukuje obě dvě metody stejně. V případě požáru dál od výztuže nebo požáru z obou stran se metody rozcházejí v řádu pár stovek Nm, to je dáno tím, že každá metoda jinak započítává únosnost betonu. Z tabulky je také patrné, že čím je delší doba trvání požáru, tím se únosnosti výrazně snižují a je velmi důležité, kde je umístěna výztuž vzhledem k uvažovanému požáru.

Tímto srovnáním jsem si ověřila správnost mých kódů a také jsem dokázala, že tyto metody, ač mají jinou úvahu výpočtu, dávají dosti podobné výsledky.

5 Shrnutí

V teoretické části této diplomové práce jsem detailně popsala vstupní údaje pro následnou praktickou část.

Věnovala jsem se popisu teplotních křivek, teorii sdílení tepla, materiálovým charakteristikám betonu, numerickým metodám, přesněji řečeno metodě konečných prvků a rovnicím vedení tepla. Díky tomu jsem byla schopná ve druhé části této práce sestrojít kód pro přestup tepla v železobetonové konstrukci.

Dále jsem se věnovala popisu proužkové metody, zónové metody a metody izotermy, díky kterým jsem, ve druhé části této práce, byla schopná sestrojít kódy k posouzení požární odolnosti.

V poslední řadě jsem se věnovala popisu již dostupných programů, které řeší podobnou problematiku a díky kterým jsem byla schopná ověřit správnost některých mých výsledků. Také jsem trochu přiblížila programovací prostředí, ve kterém jsem tvořila své kódy.

Praktickou část jsem rozdělila do tří bloků.

V prvním bloku jsem řešila modelování kódu pro 1D přestup tepla v betonové konstrukci.

Ve druhém bloku jsem modelovala proužkovou metodu a zkoumala, jaké výsledky dostanu, budu-li moment únosnosti počítat přesnějším výpočtem. V tomto bloku jsem zjistila, že zjednodušený výpočet dává vyšší únosnost než přesnější výpočet proužkovou metodou.

V posledním, tedy třetím bloku jsem modelovala metodu izotermy a zónovou metodu, které slouží pro výpočet momentu únosnosti za požární situace a provedla jsem srovnání těchto dvou metod. V rámci porovnání jsem zjistila že metody dávají dosti podobné výsledky a nelze jednoznačně říct, která metoda je přesnější. Proto pokud budu mít konstrukci, která lze počítat oběma způsoby, není příliš podstatné, který výpočet si zvolím. Také jsem zjistila že pro zónovou metodu stačí volit, normou dané, 3 proužky, protože výpočet je dosti přesný, jak jsem zjistila, tak mezi výpočtem se třemi proužky a tisíci proužky je rozdíl nepatrný.

6 Závěr

Hlavním cílem této práce bylo vytvoření výpočetního kódu pro posouzení požární odolnosti železobetonové desky podle vybraných metod. Dalším cílem byl popis matematického modelu šíření tepla, popis metody prostorové a časové diskretizace, popis numerické metody a okrajových podmínek a popis vybraných metod.

Všechny cíle této práce byly úspěšně splněny. Hlavní cíl byl naplněn vytvořením kódů v prostředí programovacího jazyka Python. Vytvořené kódy je možné použít pro zjištění průběhu teploty v železobetonové konstrukci při 1D přestupu tepla, zjištění momentu únosnosti pomocí proužkové metody a pro zjištění momentu únosnosti žb desky za požáru pomocí zónové metody a metody izotermy.

Ačkoli byly všechny cíle práce splněny a problematika byla podrobně popsána, je zde stále prostor pro další rozšíření práce. Z teoretického hlediska by bylo možné rozšíření o chování dalších materiálů za požární situace. Z praktického hlediska by bylo možné kód pro přestup tepla rozšířit o jiné materiály, popřípadě upravit na 2D přestup tepla. Kódy pro výpočet momentu únosnosti za požární situace by bylo možné upravit tak, aby zvládly počítat moment i pro prvky vystavené požáru z více než 2 stran. Dále by bylo možné z vytvořených kódů sestavit program, který by byl volně přístupný na internetu a měl by vhodnější uživatelské rozhraní než tyto moje kódy.

Přínos vidím hlavně v detailním popsání veškeré problematiky, jak v teoretické, tak v praktické části. Problematika z teoretické části by mohla sloužit jako podklad pro ruční výpočty nebo pro pochopení zmíněné látky. Problematika z praktické části by mohla sloužit jako podklad pro tvorbu podobných kódů.

Seznam obrázků

Obr. 1 - Ukázka zadávání vstupních údajů	1
Obr. 2 - Ukázka použitých vzorečků.....	2
Obr. 3 - Ukázka výpočetní tabulky	2
Obr. 4 - Výstup - rozložení teploty v čase	2
Obr. 5 - Nominální teplotní křivky[2].....	5
Obr. 6 - Jednosměrné vedení tepla (převzato z [4]).....	11
Obr. 7 - Měrné teplo v závislosti na teplotě pro jednotlivé vlhkosti [5].....	15
Obr. 8 - Změna horní (Upper) a spodní (Lower) meze tepelné vodivosti s teplotou [5].....	16
Obr. 9 - Náhrada derivací diferencemi (Vytvořeno v AutoCADu)	17
Obr. 10 - Prostorová diskretizace v 1D.....	19
Obr. 11 - Pomocný obrázek pro ukázkou výpočtu přestupu tepla v konstrukci	20
Obr. 12 - Zjednodušené rozložení napětí v nosníku (Vytvořeno v AutoCADu).....	23
Obr. 13 - Přesnější rozložení napětí v nosníku (Vytvořeno v AutoCADu).....	23
Obr. 14 - Proužková metoda, zjišťování osy x (Vytvořeno v AutoCADu)	25
Obr. 15 - Redukce průřezu desek a stěn při použití metody izotermy 500 °C (Vytvořeno v AutoCADu podle [9])	27
Obr. 16 - Redukce průřezu nosníků a sloupů při použití izotermy 500 °C (Vytvořeno v AutoCADu dle [9])	27
Obr. 17 - Součinitel $k_s(\Theta)$ pro redukci charakteristické pevnosti (f_{yk}) tahové a tlakové výztuže (třída N)[5]	28
Obr. 18 - Rozdělení napětí v obdélníkovém průřezu při požární situaci (Vytvořeno v AutoCADu dle [9])	30
Obr. 19 - Redukce průřezu při použití zónové metody ((Vytvořeno v AutoCADu dle [9])....	33
Obr. 20 - Rozdělení průřezu na zóny, stanovení teplot a redukčních součinitelů (Vytvořeno v AutoCADu dle [9])	34
Obr. 21 - Součinitel $k_{c\Theta}$ pro redukci charakteristické hodnoty pevnosti betonu v tlaku při zvýšených teplotách [5].....	35
Obr. 22 - Vstupní data programu TeDisOn [1P].....	38
Obr. 23 - Vstupní údaje z programu RCCfi [2P].....	39
Obr. 24 - Vstupní údaje programu TempAnalysis [3P].....	41
Obr. 25 - Nominální teplotní křivky (Vytvořeno v prostředí Python podle [1])	44
Obr. 26 - Nominální teplotní křivky -graf (Vytvořeno v prostředí Python podle [1]).....	44

Obr. 27 - Měrné teplo - Beton suchý (Vytvořeno v prostředí Python podle [5]).....	45
Obr. 28 - Měrné teplo - Beton s 1,5 a 3 % vlhkostí (Vytvořeno v prostředí Python podle [5])	45
Obr. 29 - Měrné teplo - graf (Vytvořeno v prostředí Python podle [5])	45
Obr. 30 – Objemová hmotnost (Vytvořeno v prostředí Python podle [5])	46
Obr. 31 – Objemová hmotnost - graf (Vytvořeno v prostředí Python podle [5])	46
Obr. 32 - Součinitel tepelné vodivosti pro beton (Vytvořeno v prostředí Python podle [5]) ..	47
Obr. 33 - Graf součinitele tepelné vodivosti betonu v závislosti na teplotě (Vytvořeno v prostředí Python podle [5])	47
Obr. 34- Derivace součinitele tepelné vodivosti pro beton (Vytvořeno v prostředí Python podle [5]).....	48
Obr. 35- Graf derivace součinitele tepelné vodivosti betonu v závislosti na teplotě (Vytvořeno v prostředí Python podle [5])	48
Obr. 36 - Pomocná tabulka pro názornou ukázkou principu příkazů	49
Obr. 37 - Příkaz pro proměnnou teplotu - vytvořeno v prostředí Python	50
Obr. 38 - Příkaz pro proměnnou teplotu 2 - vytvořeno v prostředí Python	50
Obr. 39 - Načtení knihoven a pomocných souborů (Vytvořeno v prostředí Python)	50
Obr. 40 - Vstupní údaje (Vytvořeno v prostředí Python).....	51
Obr. 41 - Okrajové a počáteční podmínky (Vytvořeno v prostředí Python).....	51
Obr. 42 - Teploty pro první řádek (Vytvořeno v prostředí Python).....	52
Obr. 43 - Vstupy pro parametrickou křivku zadané přímo (Vytvořeno v prostředí Python podle [1]).....	52
Obr. 44- Vstupy pro parametrickou křivku zadané základními údaji (Vytvořeno v prostředí Python podle [1])	53
Obr. 45 - Tvorba kódu 1. část (Vytvořeno v prostředí Python).....	54
Obr. 46- Tvorba kódu 2. část (Vytvořeno v prostředí Python)	54
Obr. 47- Tvorba kódu 3. část (Vytvořeno v prostředí Python)	55
Obr. 48 - Tvorba kódu - vykreslení (Vytvořeno v prostředí Python)	55
Obr. 49 - Ukázka možného rozložení teploty při normové nominální křivce (Vytvořeno v prostředí Python).....	56
Obr. 50 - Ukázka možného rozložení teploty při parametrické křivce (Vytvořeno v prostředí Python).....	56
Obr. 51 - Ukázka části výsledků možného rozložení teploty (Vytvořeno v prostředí Python)	56
Obr. 52 - TeDisOn - Průběh teploty v konstrukci - požár z jedné strany [1P].....	58
Obr. 53 - FiDeS - Průběh teploty v konstrukci - požár z jedné strany [3P].....	58
Obr. 54 - Python - Průběh teploty v konstrukci - požár z jedné strany	58

Obr. 55 - Průběh teplot v konstrukci - požár z jedné strany	59
Obr. 56 - TeDisOn - Průběh teploty v konstrukci - požár ze 2 stran [1P]	60
Obr. 57 - FiDeS - Průběh teploty v konstrukci - požár ze 2 stran [3P].....	60
Obr. 58 - Python - Průběh teploty v konstrukci - požár ze 2 stran.....	60
Obr. 59 - Průběh teplot v konstrukci - požár ze 2 stran	61
Obr. 60 - Parabolicko_rektangulární pracovní diagram betonu ((Vytvořeno v prostředí Python podle [11]).....	63
Obr. 61 - Bilineární pracovní diagram betonu (Vytvořeno v prostředí Python podle [11])	64
Obr. 62 - Pracovní diagram betonu - graf (Vytvořeno v prostředí Python podle [11])	64
Obr. 63 - Pracovní diagram betonu za požáru - funkce (Vytvořeno v prostředí Python podle [5]).....	65
Obr. 64 - Pracovní diagram betonu za požáru – kód pro vykreslení (Vytvořeno v prostředí Python podle [5])	65
Obr. 65 - Pracovní diagram betonu za požáru - graf (Vytvořeno v prostředí Python podle [5])65	65
Obr. 66 - Součinitel k_s pro beton s křemičitým kamenivem (k_1)/ s vápencovým kamenivem (k_2) (Vytvořeno v prostředí Python podle [5])	66
Obr. 67 - Součinitel k_s pro redukci pevnosti betonu – graf závislosti k_s na teplotě (Vytvořeno v prostředí Python podle [5])	66
Obr. 68 - Přetvoření betonu ϵ_c (Vytvořeno v prostředí Python podle [5]).....	67
Obr. 69 – Mezní přetvoření betonu ϵ_{cu} (Vytvořeno v prostředí Python podle [5])	67
Obr. 70 - Graf závislosti přetvoření na teplotě (Vytvořeno v prostředí Python podle [5]).....	68
Obr. 71 - Pracovní diagram oceli (Vytvořeno v prostředí Python podle [11])	69
Obr. 72 - Pracovní diagram oceli - graf (Vytvořeno v prostředí Python podle [11])	69
Obr. 73 - Pracovní diagram oceli za požáru (Vytvořeno v prostředí Python podle [5]).....	70
Obr. 74 - Pracovní diagram oceli za požáru – graf ((Vytvořeno v prostředí Python podle [5])70	70
Obr. 75 - Součinitel k_s pro tahovou výztuž válcovanou za tepla (k_1) a za studena (k_2) (Vytvořeno v prostředí Python podle [5]).....	71
Obr. 76 - Součinitel k_s pro tlakovou a tahovou výztuž (Vytvořeno v prostředí Python podle [5]).....	71
Obr. 77 - Součinitel k_s pro ocel – graf závislosti k_s na teplotě (Vytvořeno v prostředí Python podle [5]).....	71
Obr. 78 - Poměr meze úměrnosti a pevnosti oceli (Vytvořeno v prostředí Python podle [5])	72
Obr. 79 - Graf poměru meze únosnosti a pevnosti oceli v závislosti na teplotě (Vytvořeno v prostředí Python podle [5])	72

Obr. 80 - Poměr sklonu v lineárně pružné oblasti s modulem pružnosti (Vytvořeno v prostředí Python podle [5])	73
Obr. 81 - Graf poměru sklonu v lineárně pružné oblasti s modulem pružnosti v závislosti na teplotě (Vytvořeno v Prostředí python podle [5])	73
Obr. 82 - Vstupní údaje pro proužkovou metodu (Vytvořeno v prostředí Python)	74
Obr. 83 - Modelování proužkové metody (Vytvořeno v prostředí Python)	74
Obr. 84 - Určení mezního přetvoření v pracovním diagramu betonu (Vytvořeno v prostředí Python)	77
Obr. 85 - Rozložení napětí pro různé mezní přetvoření (Vytvořeno v AutoCADu)	78
Obr. 86 - Určení uvažovaného rozpětí přetvoření v pracovním diagramu oceli (Vytvořeno v prostředí Python)	78
Obr. 87 - Rozložení uvažovaného přetvoření (Vytvořeno v AutoCADu)	78
Obr. 88 - Vstupní údaje pro proužkovou metodu za požáru (Vytvořeno v prostředí Python). 79	
Obr. 89 - Interpolace teplot (Vytvořeno v prostředí Python)	79
Obr. 90 - Proužková metoda za požáru - přetvoření, součinitel k_s , moment únosnosti (Vytvořeno v prostředí Python)	80
Obr. 91 - Proužková metoda za požáru - výpočet (Vytvořeno v prostředí Python)	81
Obr. 92 - Výstup z proužkové metody za požáru (Vytvořeno v prostředí Python)	81
Obr. 93 - Rozložení napětí v průřezu (Vytvořeno v AutoCADu)	84
Obr. 94 - Vstupní údaje - rozměry a materiály (Vytvořeno v prostředí Python)	86
Obr. 95 - Zadání požáru - teploty převzaty z mého kódu pro přestup tepla (Vytvořeno v prostředí Python)	86
Obr. 96 - Charakteristika požáru (Vytvořeno v prostředí Python)	87
Obr. 97 - Součinitel k_s (Vytvořeno v prostředí Python podle [5])	87
Obr. 98 - Výpočet síly a momentu (Vytvořeno v prostředí Python podle [5])	88
Obr. 99 - Výsledky z mého kódu (Vytvořeno v prostředí Python)	88
Obr. 100 - Vstupní údaje - rozměry a materiály (Vytvořeno v prostředí Python)	89
Obr. 101 - Určení požáru a počtu zón (Vytvořeno v prostředí Python)	89
Obr. 102 - Rozložení teploty v průřezu - převzato z mého kódu pro přestup tepla v konstrukci (Vytvořeno v prostředí Python)	90
Obr. 103 - Určení teplot v jednotlivých zónách (Vytvořeno v prostředí Python podle [5])	90
Obr. 104 - Určení pevnosti prutů (Vytvořeno v prostředí Python podle [5])	91
Obr. 105 - Výpočet tlacení oblasti, síly ve výztuži a momentu únosnosti (Vytvořeno v prostředí Python podle [5])	91
Obr. 106 - Výsledky z mého kódu (Vytvořeno v prostředí Python)	92

Seznam tabulek

Tab. 1 - Minimální šířka průřezu	26
Tab. 2 – Součinitel $k_s(\Theta)$ pro redukci charakteristické pevnosti výztuže (Třída N) v číslech – převzato z [5].....	29
Tab. 3 - Součinitel pro redukci charakteristické hodnoty pevnosti v tlaku při zvýšených teplotách v číslech (převzato z [5])	35
Tab. 4 - Porovnání výsledků - Síly	75
Tab. 5 - Porovnání výsledků - Momenty	76
Tab. 6 - Porovnání momentů únosnosti za požáru pomocí proužkové metody	83
Tab. 7 - Porovnání podrobnosti výpočtu zónové metody	92
Tab. 8 - Porovnání výstupů ze zónové metody a z metody izotermy	93

Literatura

- [1] ČSN EN 1991-1-2: Eurokód 1: Zatížení konstrukcí - Část 1-2: Obecná zatížení - Zatížení konstrukcí vystavených účinkům požáru. Praha: Český normalizační institut, 2004.
- [2] Tabulky: Nominální teplotní křivky: SD007a-CZ-EU [online]. [vid. 2020-03-21]. Dostupné z: http://steel.fsv.cvut.cz/Access_Steel_CZ/SD_Tabulky/SD007a-CZ-EU.pdf
- [3] VOLF, I., JAREŠOVÁ, M. a OUHRABKA, M.. Přenos tepla: Studijní text pro řešitele FO a ostatní zájemce o fyziku [online]. [cit. 2020.10.09]. Dostupné z: <http://fyzikalniolympiada.cz/texty/texttz.pdf>
- [4] Sdílení tepla a úspory zateplením - I. [online]. In: . Fakulta strojní , Technická univerzita Liberec, 2011 [cit. 2020-10-09]. Dostupné z: http://kke.zcu.cz/export/sites/kke/old_web/_files/projekty/enazp/12/MMT/062_Sdileni_teplo_a_a_uspory_zateplenim_-_P2.pdf. nedatováno.
- [5] ČSN EN 1992-1-2: Eurokód 2: Navrhování betonových konstrukcí - Část 1-2: Obecná pravidla - Navrhování konstrukcí na účinky požáru. Praha: Český normalizační institut, 2006.
- [6] Mašín, D. Numerické metody [online]. In: . UK - Přírodovědecká fakulta [cit. 2020-10-10]. Dostupné z: <https://web.natur.cuni.cz/uhigug/masin/vyuka/download/prednaska3-Geotech-handouts.pdf>. nedatováno.
- [7] FUSEK, M. a HALAMA, R. MKP a MHP - interaktivní studijní materiál [online]. In: . VŠB-TU Ostrava, Západočeská univerzita v Plzni [cit. 2020-10-10]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/mkpamhp_obr.pdf. nedatováno.
- [8] Přednášky z předmětu 134NKPZ - Nosné konstrukce za požáru, Fakulta stavební, ČVUT. In: . B.m.
- [9] PROCHÁZKA, J., ŠTEFAN, R. a VAŠKOVÁ, J.. Navrhování betonových a zděných konstrukcí na účinky požáru. V Praze: České vysoké učení technické, 2010. ISBN 978-80-01-04613-5. nedatováno.
- [10] HOLAN, J. A ŠTEFAN, R. Numerická analýza rozložení teploty v betonové desce při požáru pomocí webového řešení. B.m.: 26. mezinárodní konferenční dny betonu, 2019. Fenomény pevných látek.
- [11] ČSN EN 1992-1-1: Eurokód 2: Navrhování betonových konstrukcí - Část 1-1: Obecná pravidla a pravidla pro pozemní stavby. Praha: Český normalizační institut, 2006.

Programy

- [1P] TeDisOn - Temperature Distribution Online [program, online]. [vid. 2020-04-20]. Dostupné z: <http://people.fsv.cvut.cz/~holanjak/software/tedison/program/>
- [2P] RCCfi - Výpočetní program pro posouzení požární odolnosti železobetonových sloupů [program]. [vid. 2020-10-09]. Dostupné z: <http://people.fsv.cvut.cz/www/stefarad/software/rccfi/rccfi.cz.html>
- [3P] FiDeS - Soubor výpočetních programů pro navrhování betonových a zděných konstrukcí na účinky požáru podle Eurokódů [program]. [vid. 2020-10-09]. Dostupné z: <http://people.fsv.cvut.cz/www/stefarad/software/fides/fides.html>
- [4P] Ansys [software]. [vid. 2020-04-20]. Dostupné z: <https://www.techsoft-eng.cz/software/>
- [5P] Python vs Matlab [online]. In: The Pyzo team [cit. 2020-10-14]. Dostupné z: https://pyzo.org/python_vs_matlab.html