

# Řešení 1D teplotní rovnice metodou sítí

## Modelování přenosu tepla v geologii

---

Ondrej Lexa

2017

Ústav petrologie a strukturní geologie

Přírodovědecká fakulta UK v Praze

Albertov 6

`lexa@natur.cuni.cz`

# Table of contents

1. Metoda sítí
2. Stacionární řešení rovnice vedení tepla
3. Evoluční řešení rovnice vedení tepla
4. Vedení tepla s variabilní konduktivitou a produkcí tepla

## Metoda sítí

---

# Metoda sítí neboli metoda konečných diferencí

Spolu s metodou konečných prvků jedna ze dvou základních numerických metod pro přibližné řešení diferenciálních rovnic na počítači. Metoda sítí spočívá v tom, že v oblasti, ve které se hledá řešení dané diferenciální rovnice, se zvolí konečný systém bodů (sítě) a v každém z nich se nahradí v diferenciální rovnici derivace hledané funkce diferencemi používajícími pouze funkční hodnoty v bodech sítě. Původní úlohu tak lze nahradit soustavou konečně mnoha (obecně nelineárních) algebraických rovnic, jejíž řešení se považuje za přibližné hodnoty skutečného řešení původní úlohy v bodech sítě. Pro celou řadu okrajových úloh je uspokojivě vyřešena otázka přesnosti takto získané aproximace (odhady nepřesnosti, věty o konvergenci).

# Aproximace funkcí I

Za určitých předpokladů o funkci  $f(x)$  v okolí bodu  $a$  lze tuto funkci vyjádřit (rozvinout) jako mocninnou řadu. Toto vyjádření funkce prostřednictvím **Taylorovy řady** se označuje jako **Taylorův rozvoj**.

## Taylorův rozvoj

$$\begin{aligned} f(x) &= f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots \\ &= \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k \end{aligned} \tag{1}$$

Pro přibližné vyjádření hodnot funkce není nutné vyjadřovat všechny členy Taylorovy řady, ale můžeme zanedbat členy s vyššími derivacemi. Získáme tím tzv. **Taylorův polynom**, který aproximuje hodnoty funkce, která má v daném bodě derivaci, pomocí polynomu, jehož koeficienty závisí na derivacích funkce v tomto bodě.

## Aproximace funkcí II

Pro pravé okolí bodu je možné rovnici (1) napsat jako

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} \Delta x^2 + \frac{f^{(3)}(x)}{3!} \Delta x^3 + \dots \quad (2)$$

Pro levé okolí bodu je možné rovnici (1) napsat jako

$$f(x - \Delta x) = f(x) - \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} \Delta x^2 - \frac{f^{(3)}(x)}{3!} \Delta x^3 + \dots \quad (3)$$

# Aproximace první derivace zprava

Řešením rovnice (2)

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} \Delta x^2 + \frac{f^{(3)}(x)}{3!} \Delta x^3 + \dots$$

pro  $f'(x)$  dostaneme

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{f''(x)}{2!} \Delta x - \frac{f^{(3)}(x)}{3!} \Delta x^2 - \dots$$

Pro výpočet konečné difference první derivace zprava použijeme

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x) \quad (\Delta x \rightarrow 0)$$

# Aproximace první derivace zleva

Řešením rovnice (3)

$$f(x - \Delta x) = f(x) - \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} \Delta x^2 - \frac{f^{(3)}(x)}{3!} \Delta x^3 + \dots$$

pro  $f'(x)$  dostaneme

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + \frac{f''(x)}{2!} \Delta x - \frac{f^{(3)}(x)}{3!} \Delta x^2 + \dots$$

Pro výpočet konečné difference první derivace zleva použijeme

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + O(\Delta x) \quad (\Delta x \rightarrow 0)$$



# Centrální aproximace první derivace

Odečtením rovnice (3) od rovnice (2) dostaneme

$$f(x + \Delta x) - f(x - \Delta x) = 2\frac{f'(x)}{1!}\Delta x + 2\frac{f^{(3)}(x)}{3!}\Delta x^3 + 2\frac{f^{(5)}(x)}{5!}\Delta x^5 + \dots$$

Řešením pro  $f'(x)$  dostaneme

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} - \frac{f^{(3)}(x)}{3!}\Delta x^2 + \frac{f^{(5)}(x)}{5!}\Delta x^4 + \dots$$

**Pro výpočet centrální konečné diference první derivace použijeme**

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O(\Delta x^2) \quad (\Delta x \rightarrow 0)$$

# Aproximace druhé derivace

Sečtením rovnice (3) od rovnice (2) dostaneme

$$f(x+\Delta x)+f(x-\Delta x) = 2f(x)+2\frac{f''(x)}{2!}\Delta x^2+2\frac{f^{(4)}(x)}{4!}\Delta x^4+2\frac{f^{(6)}(x)}{6!}\Delta x^6+\dots$$

Řešením pro  $f''(x)$  dostaneme

$$f''(x) = \frac{f(x-\Delta x) - 2f(x) + f(x+\Delta x)}{\Delta x^2} + 2\frac{f^{(4)}(x)}{4!}\Delta x^2 + 2\frac{f^{(6)}(x)}{6!}\Delta x^4 + \dots$$

Pro výpočet konečné difference druhé derivace použijeme

$$f''(x) = \frac{f(x-\Delta x) - 2f(x) + f(x+\Delta x)}{\Delta x^2} + O(\Delta x^2) \quad (\Delta x \rightarrow 0)$$

# Stacionární řešení rovnice vedení tepla

---

# Rovnice vedení tepla

Pro případ konstantní konduktivity je možné rovnici vedení tepla upravit do tvaru

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} + \frac{H}{\rho \cdot c}$$

kde

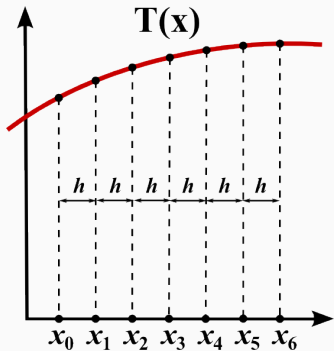
$$\kappa = \frac{\lambda}{\rho \cdot c}$$

je **tepelná difuzivita** (součinitel teplotní vodivosti).

Stav, kdy teplotní rozdíl mezi jednotlivými částmi tělesa se v čase nemění t.j.  $\frac{\partial T}{\partial t} = 0$ , nazýváme stacionárním vedením tepla. Rovnice vedení tepla má tvar **Poissonovy rovnice**

$$\frac{\partial^2 T}{\partial x^2} = -\frac{H}{\lambda} \tag{4}$$

# Numerické řešení Poissonovy rovnice I



Aproximaci řešení rovnice (4) získáme nahrazením druhé derivace konečnou diferencí druhé derivace

$$f''(x) \approx \frac{f(x - \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2}$$

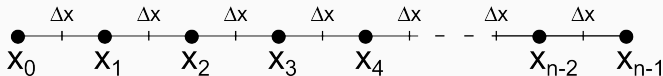
pro všechny  $x_i$  pro  $i = 1 \dots n - 2$

Pro body  $x_0$  a  $x_{n-1}$  použijeme rovnice odvozené z okrajových podmínek.

Pro každý diskrétní bod  $x_i$  pro  $i = 1 \dots n - 2$  tak platí

$$\frac{T_{x_i - \Delta x} - 2T_{x_i} + T_{x_i + \Delta x}}{\Delta x^2} = \frac{-H}{\lambda}$$

## Numerické řešení Poissonovy rovnice II



Dostaneme tak soustavu  $n - 2$  rovnic

$$T_{x_0} - 2T_{x_1} + T_{x_2} = -\frac{H \cdot \Delta x^2}{\lambda}$$

$$T_{x_1} - 2T_{x_2} + T_{x_3} = -\frac{H \cdot \Delta x^2}{\lambda}$$

$$T_{x_2} - 2T_{x_3} + T_{x_4} = -\frac{H \cdot \Delta x^2}{\lambda}$$

$\vdots$

$$T_{x_{n-3}} - 2T_{x_{n-2}} + T_{x_{n-1}} = -\frac{H \cdot \Delta x^2}{\lambda}$$

# Okrajové podmínky – Dirichletova okrajová podmínka

Řešením rovnice vedení tepla je rozložení teplot v prostoru a čase za pomocí **okrajových** a u nestacionárních úloh i **počátečních** podmínek. Okrajové podmínky určují rozložení teploty na povrchu tělesa. Pro 1D úlohy potřebujeme definovat teplotu pro body  $x_0$  a  $x_{n-1}$ .

**Dirichletova podmínka** - předepsaná hodnota teplota okrajových bodů

$$T_{x_0} = T(t)$$

$$T_{x_{n-1}} = T(t)$$

$$T_{x_0} = T_{def}$$

$$T_{x_{n-1}} = T_{def}$$

# Numerické řešení Poissonovy rovnice III

Ukázka soustavy  $n$  lineárních rovnic o  $n$  neznámých pro řešení Poissonovy rovnice a použití Dirichletovy podmínky pro okrajové body. Dostáváme tak soustavu

$$\begin{aligned}T_{x_0} &= T_{def} \\T_{x_0} - 2T_{x_1} + T_{x_2} &= -\frac{H \cdot \Delta x^2}{\lambda} \\T_{x_1} - 2T_{x_2} + T_{x_3} &= -\frac{H \cdot \Delta x^2}{\lambda} \\T_{x_2} - 2T_{x_3} + T_{x_4} &= -\frac{H \cdot \Delta x^2}{\lambda} \\\vdots \\T_{x_{n-3}} - 2T_{x_{n-2}} + T_{x_{n-1}} &= -\frac{H \cdot \Delta x^2}{\lambda} \\T_{x_{n-1}} &= T_{def}\end{aligned}$$



# Numerické řešení Poissonovy rovnice IV

v maticovém tvaru lze soustavu rovnic zapsat

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} T_{x_0} \\ T_{x_1} \\ T_{x_2} \\ \vdots \\ T_{x_{n-2}} \\ T_{x_{n-1}} \end{vmatrix} = \begin{vmatrix} T_{def} \\ -\frac{H \cdot \Delta x^2}{\lambda} \\ -\frac{H \cdot \Delta x^2}{\lambda} \\ \vdots \\ -\frac{H \cdot \Delta x^2}{\lambda} \\ T_{def} \end{vmatrix}$$

kde matice soustavy je čtvercová regulární matice, tudíž soustava rovnic má právě jedno řešení. Zkráceně lze systém zapsat

$$\mathbf{A}x = b$$

## Ukázka kódu v Pythonu

```
# Stacionarní řešení 1D Poissonovy rovnice
# =====

# Import potřebných modulů
from pylab import *
from scipy.sparse import spdiags
from scipy.sparse.linalg import spsolve

# Parametry: konduktivita, produktivita,
# počet uzlu, velikost domeny
k, H, n, tc = 2.5, 1e-6, 100, 35000

# inicializace
x = linspace(0, tc, n) # pozice uzlu
dx = tc / (n - 1)      # vzdálenost uzlu
d = ones(n)
```

## Ukázka kódu v Pythonu - pokračování

```
# matice soustavy
A = spdiags([d, -2*d, d], [-1, 0, 1], n, n, 'csr')
# prava strana soustavy
b = -d * H * dx**2 / k

# Dirichletova okrajova podmínka T0
A[0, :2] = [1, 0]
b[0] = 0
# Dirichletova okrajova podmínka Tn-1
A[-1, -2:] = [0, 1]
b[-1] = 100
```

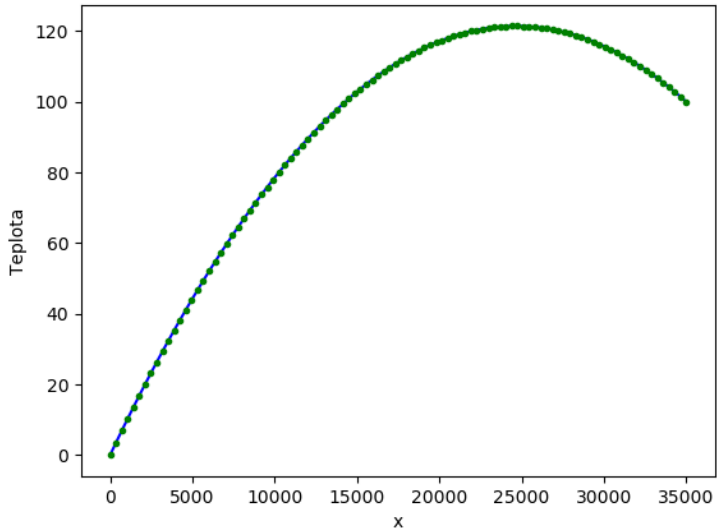
## Ukázka kódu v Pythonu - pokračování

```
# Reseni soustavy
t = spsolve(A, b)

# Analyticke reseni
C1 = 100 / tc + H * tc / (2 * k)
C2 = 0
ta = -H * x**2 / (2 * k) + C1 * x + C2

# Vykresli numericke a analyticke reseni
plot(x, t, 'b', x, ta, 'g.')
xlabel('x')
ylabel('Teplota')

# zabrani zavreni obrazku po ukonceni skriptu
show()
```



# Okrajové podmínky – Neumannova okrajová podmínka I

Neumannova podmínka - předepsaná hustota tepelného toku

$$q = -\lambda \cdot \frac{\partial T}{\partial x}$$

Jednou z možností je použít aproximace první derivace zprava nebo zleva podle pozice. **Pozor! Tyto aproximace mají přesnost  $O(\Delta x)$ .**

$$q = -\lambda \frac{T_{x_1} - T_{x_0}}{\Delta x}$$

$$T_{x_0} - T_{x_1} = \frac{q \cdot \Delta x}{\lambda}$$

$$q = -\lambda \frac{T_{x_{n-1}} - T_{x_{n-2}}}{\Delta x}$$

$$-T_{x_{n-2}} + T_{x_{n-1}} = -\frac{q \cdot \Delta x}{\lambda}$$

## Okrajové podmínky – Neumannova okrajová podmínka II

Výhodnější je pro definování Neumannovi okrajové podmínky použít centrální diferenci s přesností  $O(\Delta x^2)$  pomocí tzv. *fiktivních uzlů* ležících mimo výpočetní doménu ve vzdálenosti  $\Delta x$ .

$$q = -\lambda \frac{T_{x_1} - T_F}{2\Delta x}$$

$$T_F = \frac{q \cdot 2\Delta x}{\lambda} + T_{x_1}$$

$$q = -\lambda \frac{T_F - T_{x_{n-2}}}{2\Delta x}$$

$$T_F = -\frac{q \cdot 2\Delta x}{\lambda} + T_{x_{n-2}}$$

Takto vyjádřené hodnoty teploty ve *fiktivních uzlech* můžeme substitovat do aproximace rovnice vedení tepla napsané pro okrajové uzly.

$$T_F - 2T_{x_0} + T_{x_1} = -\frac{H \cdot \Delta x^2}{\lambda}$$

$$T_{x_{n-2}} - 2T_{x_{n-1}} + T_F = -\frac{H \cdot \Delta x^2}{\lambda}$$

$$-2T_{x_0} + 2T_{x_1} = -\frac{q \cdot 2\Delta x + H \cdot \Delta x^2}{\lambda}$$

$$2T_{x_{n-2}} - 2T_{x_{n-1}} = \frac{q \cdot 2\Delta x - H \cdot \Delta x^2}{\lambda}$$

# Výpočet stabilní geotermie I

Pro výpočet stabilní geotermie můžeme použít např. Dirichletovu okrajovou podmínku na povrchu a Neumannovu okrajovou podmínku v hloubce. Dostáváme tak soustavu  $n$  rovnic o  $n$  neznámých:

$$\begin{aligned}T_{x_0} &= T_{def} \\T_{x_0} - 2T_{x_1} + T_{x_2} &= -\frac{H \cdot \Delta x^2}{\lambda} \\T_{x_1} - 2T_{x_2} + T_{x_3} &= -\frac{H \cdot \Delta x^2}{\lambda} \\T_{x_2} - 2T_{x_3} + T_{x_4} &= -\frac{H \cdot \Delta x^2}{\lambda} \\&\vdots \\T_{x_{n-3}} - 2T_{x_{n-2}} + T_{x_{n-1}} &= -\frac{H \cdot \Delta x^2}{\lambda} \\2T_{x_{n-2}} - 2T_{x_{n-1}} &= \frac{q \cdot 2\Delta x - H \cdot \Delta x^2}{\lambda}\end{aligned}$$



# Výpočet stabilní geotermie II

v maticovém tvaru lze soustavu rovnic zapsat

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix} \cdot \begin{pmatrix} T_{x_0} \\ T_{x_1} \\ T_{x_2} \\ \vdots \\ T_{x_{n-2}} \\ T_{x_{n-1}} \end{pmatrix} = \begin{pmatrix} T_{def} \\ -\frac{H \cdot \Delta x^2}{\lambda} \\ -\frac{H \cdot \Delta x^2}{\lambda} \\ \vdots \\ -\frac{H \cdot \Delta x^2}{\lambda} \\ \frac{q \cdot 2\Delta x - H \cdot \Delta x^2}{\lambda} \end{pmatrix}$$

kde matice soustavy je čtvercová regulární matice, tudíž soustava rovnic má právě jedno řešení. Zkráceně lze systém zapsat

$$\mathbf{A}x = b$$

## Ukážka kódu v Pythonu

```
# Stacionarni reseni 1D rovnice vedeni tepla
# =====

# Import potrebných modulu
from pylab import *
from scipy.sparse import spdiags
from scipy.sparse.linalg import spsolve

# Parametry: konduktivita, tepelny tok na bazi
# produktivita, mocnost, pocet uzlu
k, q, H, tc, n = 2.5, -0.02, 1e-6, 35000, 100

# inicializace
x = linspace(0, tc, n) # pozice uzlu
dx = tc / (n - 1)      # vzdalenost uzlu
```

## Ukázka kódu v Pythonu

```
# Vytvoreni ridke matice soustavy
d = ones(n)
A = spdiags([d, -2*d, d], [-1, 0, 1], n, n, 'csr')

# prava strana soustavy
b = -d * H * dx**2 / k

# Dirichletova okrajova podminka
A[0, :2] = [1, 0]
b[0] = 0

# Neumannova okrajova podminka
A[-1, -2:] = [2, -2]
b[-1] += 2 * q * dx / k
```

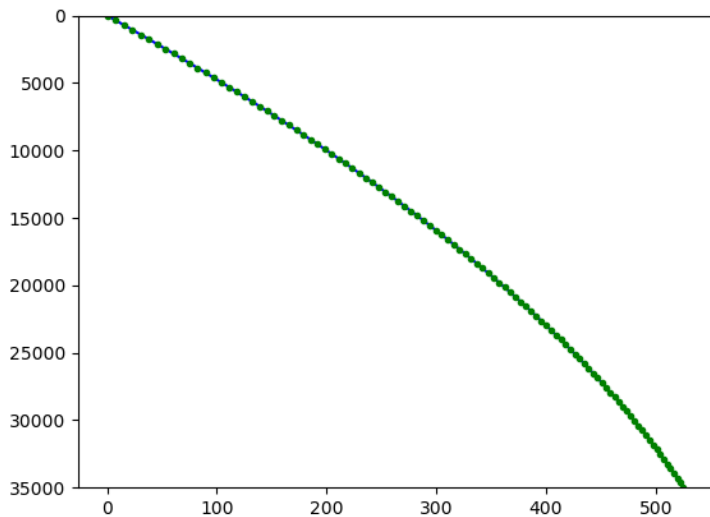
## Ukázka kódu v Pythonu

```
# Reseni soustavy
t = spsolve(A, b)

# Analyticke reseni
ta = -H*x**2/(2*k) + (tc*H/k - q/k)*x

# Vykresli numericke a analyticke reseni
plot(t, x, 'b', ta, x, 'g.')
ylim(tc, 0) # otoc osu y

# zabrani zavreni obrazku po ukonceni skriptu
show()
```



# Evoluční řešení rovnice vedení tepla

---

# Řešení evoluční rovnice vedení tepla

Pro případ konstantní konduktivity je možné rovnici vedení tepla upravit do tvaru

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} + \frac{H}{\rho \cdot c} \quad (5)$$

kde

$$\kappa = \frac{\lambda}{\rho \cdot c}$$

je **tepelná difuzivita** (součinitel teplotní vodivosti).

Stav, kdy  $\frac{\partial T}{\partial t} \neq 0$ , nazýváme nestacionárním a rovnice vedení tepla popisuje změnu teploty v prostoru a v čase a v této podobě patří do skupiny **evolučních rovnic**.

# Numerické řešení evoluční rovnice vedení tepla

Kromě prostorové derivace, nám v rovnici vystupuje první derivace teploty podle času.

Aproximaci řešení rovnice (5) získáme

- nahrazením prostorové druhé derivace konečnou diferencí druhé derivace

$$f''(x) \approx \frac{f(x - \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2}$$

pro všechny  $x_i$  pro  $i = 1 \dots n - 2$

- nahrazením první derivace podle času konečnou diferencí první derivace zprava

$$f'(x) \approx \frac{f(x)^{t+\Delta t} - f(x)^t}{\Delta t}$$

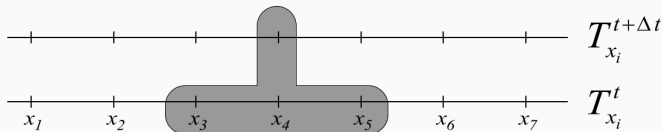
pro všechny  $x_i$  pro  $i = 1 \dots n - 2$



# Numerické řešení - schéma FTCS

Pro každý diskrétní bod  $x_i$  pro  $i = 1 \dots n - 2$  tak platí

$$\frac{T_{x_i}^{t+\Delta t} - T_{x_i}^t}{\Delta t} = \kappa \frac{T_{x_i-\Delta x}^t - 2T_{x_i}^t + T_{x_i+\Delta x}^t}{\Delta x^2} + \frac{H}{\rho \cdot c}$$



Řešením pro  $T_{x_i}^{t+\Delta t}$  dostaneme

$$T_{x_i}^{t+\Delta t} = u T_{x_i-\Delta x}^t + (1 - 2u) T_{x_i}^t + u T_{x_i+\Delta x}^t + \frac{H\Delta t}{\rho \cdot c}$$

kde  $u$  je Fourierovo číslo

$$u = \frac{\kappa \Delta t}{\Delta x^2}$$

# Numerické řešení - schéma FTCS II.

Při použití Dirichletovy okrajové podmínky platí

$$T_{x_0}^{t+\Delta t} = T_{x_0}^t = T_{def}$$

$$T_{x_{n-1}}^{t+\Delta t} = T_{x_{n-1}}^t = T_{def}$$

$$T_{x_{n-1}}^{t+\Delta t} = 2uT_{x_{n-2}}^t + (1 - 2u) T_{x_{n-1}}^t + \frac{H\Delta t}{\rho \cdot c} - \frac{u \cdot q \cdot 2\Delta x}{\lambda}$$

$$\begin{pmatrix} T_{x_0}^{t+\Delta t} \\ T_{x_1}^{t+\Delta t} \\ T_{x_2}^{t+\Delta t} \\ \vdots \\ T_{x_{n-2}}^{t+\Delta t} \\ T_{x_{n-1}}^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ u & 1-2u & u & 0 & 0 & 0 \\ 0 & u & 1-2u & u & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & u & 1-2u & u \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} T_{def} \\ T_{x_1}^t \\ T_{x_2}^t \\ \vdots \\ T_{x_{n-2}}^t \\ T_{def} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{H\Delta t}{\rho \cdot c} \\ \frac{H\Delta t}{\rho \cdot c} \\ \vdots \\ \frac{H\Delta t}{\rho \cdot c} \\ 0 \end{pmatrix}$$

$$T^{t+\Delta t} = \mathbf{A} T^t + b$$

Podmínka stability

$$u = \frac{\kappa \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

$$\Delta t \leq \frac{\Delta x^2}{2\kappa}$$

# Numerické řešení - schéma FTCS III.

Při použití Dirichletovy okrajové podmínky na povrchu a Neumannovy okrajové podmínky v hloubce dostáváme soustavu rovnic

$$T_{x_0}^{t+\Delta t} = T_{x_0}^t = T_{def} \qquad T_{x_n}^t = -\frac{q \cdot 2\Delta x}{\lambda} + T_{x_{n-2}}^t$$

$$T_{x_{n-1}}^{t+\Delta t} = 2uT_{x_{n-2}}^t + (1-2u)T_{x_{n-1}}^t + \frac{H\Delta t}{\rho \cdot c} - \frac{u \cdot q \cdot 2\Delta x}{\lambda}$$

$$\begin{pmatrix} T_{x_0}^{t+\Delta t} \\ T_{x_1}^{t+\Delta t} \\ T_{x_2}^{t+\Delta t} \\ \vdots \\ T_{x_{n-2}}^{t+\Delta t} \\ T_{x_{n-1}}^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ u & 1-2u & u & 0 & 0 & 0 \\ 0 & u & 1-2u & u & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & u & 1-2u & u \\ 0 & 0 & 0 & 0 & 2u & 1-2u \end{pmatrix} \cdot \begin{pmatrix} T_{def} \\ T_{x_1}^t \\ T_{x_2}^t \\ \vdots \\ T_{x_{n-2}}^t \\ T_{x_{n-1}}^t \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{H\Delta t}{\rho \cdot c} \\ \frac{H\Delta t}{\rho \cdot c} \\ \vdots \\ \frac{H\Delta t}{\rho \cdot c} \\ \frac{\Delta t(H\Delta x - 2q)}{\rho \cdot c \cdot \Delta x} \end{pmatrix}$$

$$T^{t+\Delta t} = \mathbf{A} T^t + b$$

## Ukázka kódu v Pythonu

```
# Evolucni reseni 1D rovnice vedeni tepla FTCS
# =====

# Import potrebných modulu
from pylab import *
from scipy.sparse import spdiags

# Parametry: konduktivita, tepelny tok na bazi
# produktivita, mocnost, pocet uzlu
k, q, H, tc, n = 2.5, -0.02, 1e-6, 35000, 100
rho, c, ysec = 2700, 900, 365.25*24*3600

# inicializace
x = linspace(0, tc, n) # pozice uzlu
dx = tc/(n-1)          # vzdalenost uzlu
kappa = k/(rho*c)
```

## Ukázka kódu v Pythonu

```
# Vytvoreni ridke matice soustavy
dt = 100*ysec
d = ones(n)
u = kappa*dt/dx**2
A = spdiags([d*u, 1 - 2*d*u, d*u],
            [-1, 0, 1], n, n, 'csr')

# prava strana soustavy
b = d*H*dt / (rho*c)

# Dirichletova okrajova podminka
A[0, :2] = [1, 0]
b[0] = 0

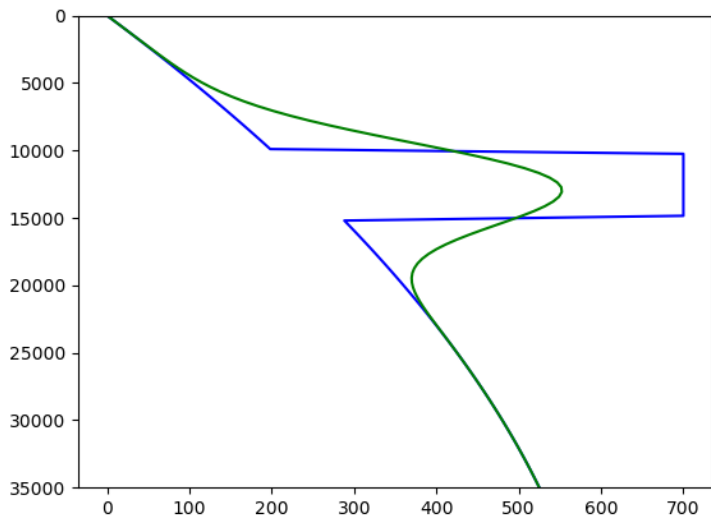
# Neumannova okrajova podminka
A[-1, -2:] = [2*u, 1-2*u]
b[-1] -= 2*dt*q / (rho*c*dx)
```

## Ukážka kódu v Pythonu

```
# Reseni vyvoje termalni perturbace za 100 tisic let
t[logical_and(x > 10000, x < 15000)] = 700
tn = t
for i in range(1000):
    tn = A.dot(tn) + b

# Vykresli inicialni a evolucionni reseni
plot(t, x, 'b', tn, x, 'g')
ylim(tc, 0) # otoc osu y

# zabrani zavreni obrazku po ukonceni skriptu
show()
```



# Numerické řešení - schéma BTCS

Po nahrazení první derivace podle času v rovnici (5) konečnou diferencí první derivace zleva

$$f'(x) \approx \frac{f(x)^t - f(x)^{t-\Delta t}}{\Delta t}$$

pro všechny  $x_i$  pro  $i = 1 \dots n-1$  pro každý diskrétní bod  $x_i$  tak platí

$$\frac{T_{x_i}^t - T_{x_i}^{t-\Delta t}}{\Delta t} = \kappa \frac{T_{x_i-\Delta x}^t - 2T_{x_i}^t + T_{x_i+\Delta x}^t}{\Delta x^2} + \frac{H}{\rho \cdot c}$$

Úpravou dostaneme

$$-uT_{x_i-\Delta x}^t + (1 + 2u)T_{x_i}^t - uT_{x_i+\Delta x}^t = \frac{H\Delta t}{\rho \cdot c} + T_{x_i}^{t-\Delta t}$$

kde  $u$  je Fourierovo číslo

$$u = \frac{\kappa \Delta t}{\Delta x^2}$$

Dostáváme tak soustavu rovnic, kterou musíme vyřešit.



# Numerické řešení - schéma BTCS II.

Pro stejné okrajové podmínky jako pro stacionární řešení platí:

$$T_{x_0}^t = T_{x_0}^{t-\Delta t} = T_{def} \qquad T_{x_n}^t = -\frac{q \cdot 2\Delta x}{\lambda} + T_{x_{n-2}}^t$$

$$-2uT_{x_{n-2}}^t + (1 + 2u) T_{x_{n-1}}^t = \frac{H\Delta t}{\rho \cdot c} - \frac{u \cdot q \cdot 2\Delta x}{\lambda} + T_{x_{n-1}}^{t-\Delta t}$$

v maticovém tvaru lze soustavu rovnic zapsat:

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -u & 1+2u & -u & 0 & 0 & 0 \\ 0 & -u & 1+2u & -u & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u & 1+2u & -u \\ 0 & 0 & 0 & 0 & -2u & 1+2u \end{vmatrix} \cdot \begin{vmatrix} T_{x_0}^t \\ T_{x_1}^t \\ T_{x_2}^t \\ \vdots \\ T_{x_{n-2}}^t \\ T_{x_{n-1}}^t \end{vmatrix} = \begin{vmatrix} T_{def} \\ \frac{H\Delta t}{\rho \cdot c} + T_{x_1}^{t-\Delta t} \\ \frac{H\Delta t}{\rho \cdot c} + T_{x_2}^{t-\Delta t} \\ \vdots \\ \frac{H\Delta t}{\rho \cdot c} + T_{x_{n-2}}^{t-\Delta t} \\ \frac{\Delta t(H\Delta x - 2q)}{\rho \cdot c \cdot \Delta x} + T_{x_{n-1}}^{t-\Delta t} \end{vmatrix}$$

kde matice soustavy je čtvercová regulární matice, tudíž soustava rovnic má právě jedno řešení. Zkráceně lze systém zapsat:

$$\mathbf{A}x = b$$

## Ukázka kódu pro evoluční řešení - funkce FTCS

```
def ftcs(t, dt, q=-0.02, T0=0):  
    # tvorba matice soustavy  
    u = kappa*dt*ysec / dx**2  
    A = spdiags([d*u, 1 - 2*d*u, d*u],  
                [-1, 0, 1], n, n, 'csr')  
    b = d*H*dt*ysec / (rho*c)  
  
    # okrajova podminka nahore  
    A[0, :2] = [1, 0]  
    b[0] = 0  
    # okrajova podminka dole  
    A[-1, -2:] = [2*u, 1 - 2*u]  
    b[-1] -= u*q*2*dx / k # 2*dt*q/(rho*c*dx)  
    # vyres  
    return A.dot(t) + b
```

## Ukázka kódu pro evoluční řešení - funkce BTCS

```
def btcs(t, dt, q=-0.02, T0=0):  
    # tvorba matice soustavy  
    u = kappa*dt*ysec / dx**2  
    A = spdiags([-d*u, 1 + 2*d*u, -d*u],  
                [-1, 0, 1], n, n, 'csr')  
    b0 = d*H*dt*ysec / (rho*c)  
  
    # okrajova podminka nahore  
    A[0, :2] = [1, 0]  
    b0[0] = 0  
  
    # okrajova podminka dole  
    A[-1, -2:] = [-2*u, 1 + 2*u]  
    b0[-1] -= u*q*2*dx / k # 2*dt*q/(rho*c*dx)  
    # vyres  
    return spsolve(A, b0 + t)
```

## Vedení tepla s variabilní konduktivitou a produkcí tepla

---

# Řešení rovnice vedení tepla s variabilní konduktivitou a produkcí tepla

Pro případ variabilní konduktivity (konduktivita je funkce) má evoluční rovnice vedení tepla tvar

$$\rho \cdot c \cdot \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left( \lambda \cdot \frac{\partial T}{\partial x} \right) + H \quad (6)$$

Stav, kdy teplotní rozdíl mezi jednotlivými částmi tělesa se v čase nemění t.j.  $\frac{\partial T}{\partial t} = 0$ , nazýváme stacionárním vedením tepla. Rovnice vedení tepla má tvar

$$\frac{\partial}{\partial x} \left( \lambda \cdot \frac{\partial T}{\partial x} \right) = -H \quad (7)$$

# Úprava rovnice vedení tepla s variabilní konduktivitou

Rovnice (6) a (7) můžeme upravit pomocí pravidla o derivaci součinu funkcí do tvaru:

$$\rho \cdot c \cdot \frac{\partial T}{\partial t} = \frac{\partial \lambda}{\partial x} \cdot \frac{\partial T}{\partial x} + \lambda \cdot \frac{\partial^2 T}{\partial x^2} + H$$

resp.

$$\frac{\partial \lambda}{\partial x} \cdot \frac{\partial T}{\partial x} + \lambda \cdot \frac{\partial^2 T}{\partial x^2} = -H$$

V rovnici vystupuje první a druhá prostorová derivace teploty a konduktivity.

# Numerické řešení stacionární rovnice vedení tepla s variabilní konduktivitou

Aproximaci řešení rovnice (7) získáme

- nahrazením druhé derivace konečnou diferencí druhé derivace

$$f''(x) \approx \frac{f(x - \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2}$$

pro všechny  $x_i$  pro  $i = 1 \dots n - 1$

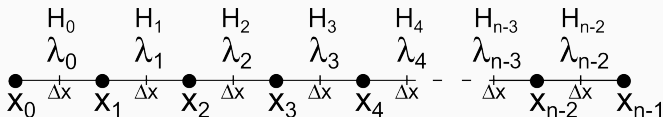
- nahrazením první derivace centrální konečnou diferencí první derivace

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

pro všechny  $x_i$  pro  $i = 1 \dots n - 1$

Pro body  $x_0$  a  $x_n$  použijeme rovnice odvozené z okrajových podmínek.

# Numerické řešení stacionární rovnice vedení tepla



V bodech  $x_i$  pro  $i = 1 \dots n - 2$  hodnotu konduktivity a produkce tepla nahradíme průměrnou hodnotou dvou nejbližších hodnot:

$$\bar{\lambda}_{x_i} = \frac{\lambda_{i-1} + \lambda_i}{2}$$
$$\bar{H}_{x_i} = \frac{H_{i-1} + H_i}{2}$$

Pro každý diskrétní bod  $x_i$  pro  $i = 1 \dots n - 2$  pak platí:

$$\frac{\lambda_i - \lambda_{i-1}}{\Delta x} \cdot \frac{T_{x_i+\Delta x} - T_{x_i-\Delta x}}{2\Delta x} + \bar{\lambda}_{x_i} \cdot \frac{T_{x_i-\Delta x} - 2T_{x_i} + T_{x_i+\Delta x}}{\Delta x^2} = -\bar{H}_{x_i}$$

$$\lambda_{i-1} T_{x_i-\Delta x} - (\lambda_{i-1} + \lambda_i) T_{x_i} + \lambda_i T_{x_i+\Delta x} = -\Delta x^2 \cdot \frac{H_{i-1} + H_i}{2}$$



# Výpočet stabilní geotermie pro variabilní konduktivitu

Pro výpočet stabilní geotermie můžeme použít např. Dirichletovu okrajovou podmínku pro  $x_1$  a Neumannovu pro  $x_{n-1}$ .

Dostáváme tak soustavu rovnic:

$$T_{x_0} = T_{def}$$

$$\lambda_0 T_{x_0} - (\lambda_0 + \lambda_1) T_{x_1} + \lambda_1 T_{x_2} = -\Delta x^2 \cdot \frac{H_0 + H_1}{2}$$

$$\lambda_1 T_{x_1} - (\lambda_1 + \lambda_2) T_{x_2} + \lambda_2 T_{x_3} = -\Delta x^2 \cdot \frac{H_1 + H_2}{2}$$

$$\lambda_2 T_{x_2} - (\lambda_2 + \lambda_3) T_{x_3} + \lambda_3 T_{x_4} = -\Delta x^2 \cdot \frac{H_2 + H_3}{2}$$

$$\vdots$$

$$\lambda_{n-3} T_{x_{n-3}} - (\lambda_{n-3} + \lambda_{n-2}) T_{x_{n-2}} + \lambda_{n-2} T_{x_{n-1}} = -\Delta x^2 \cdot \frac{H_{n-3} + H_{n-2}}{2}$$

$$2\lambda_{n-2} T_{x_{n-2}} - 2\lambda_{n-2} T_{x_{n-1}} = q \cdot 2\Delta x - H_{n-2} \cdot \Delta x^2$$

# Výpočet stabilní geotermie pro variabilní konduktivitu II

V maticovém tvaru lze soustavu rovnic zapsat  $\mathbf{A}x = b$  kde

$$\mathbf{A} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \lambda_0 & -(\lambda_0 + \lambda_1) & \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_1 & -(\lambda_1 + \lambda_2) & \lambda_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \lambda_{n-3} & -(\lambda_{n-3} + \lambda_{n-2}) & \lambda_{n-2} \\ 0 & 0 & 0 & 0 & 2\lambda_{n-2} & -2\lambda_{n-2} \end{vmatrix}$$

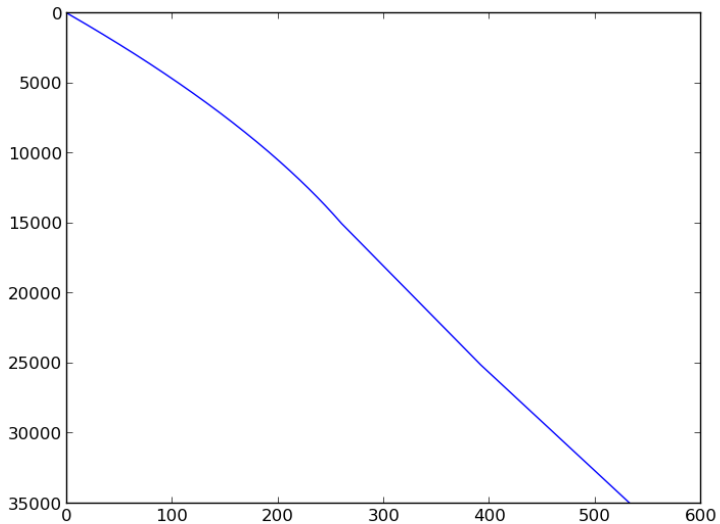
$$b = \begin{vmatrix} T_{def} \\ -\Delta x^2 \cdot \frac{H_0 + H_1}{2} \\ -\Delta x^2 \cdot \frac{H_1 + H_2}{2} \\ \vdots \\ -\Delta x^2 \cdot \frac{H_{n-3} + H_{n-2}}{2} \\ q \cdot 2\Delta x - H_{n-2} \cdot \Delta x^2 \end{vmatrix}$$

## Ukázka kódu pro stacionární řešení

```
from pylab import *
from scipy.sparse import spdiags
from scipy.sparse.linalg import spsolve
def steadystate(k, H, dx, q=-0.03, T0=0):
    n = len(k)+1
    kl,kr = k[:-1], k[1:]
    Hl,Hr = H[:-1], H[1:]
    # sestavení matice včetně okr. podmínek
    dl = hstack((kl, 2*k[-1], 0))
    dr = hstack((0, 0, kr))
    dm = hstack((1, -(kl+kr), -2*k[-1]))
    A = spdiags([dl, dm, dr], [-1, 0, 1], n, n, 'csr')
    #vektor pravy strany včetně okr. podmínek
    b = hstack((T0, -(Hl+Hr)*dx**2 / 2,
                2*q*dx - H[-1]*dx**2))
    return spsolve(A,b)
```

## Ukázka kódu pro stacionární řešení

```
dx = 100
T0, q = 0, -0.03
k = array(150*[2.6] + 100*[2.3] + 100*[2.1])
H = array(150*[2e-6] + 200*[0.0])
t = steadystate(k, H, dx, q, T0)
x = linspace(0, len(k)*dx, len(k)+1)
plot(t,x)
ylim(len(k)*dx, 0) # otoc osu y
```



## Příklad teplotně závislé konduktivity

Vypočítejte stabilní geotermu pro teplotně závislou konduktivitu danou vztahem:

$$\lambda = 2 + \exp\left(\frac{-T}{1250}\right) + \frac{48}{\exp\left(\frac{1250-T}{10}\right)}$$

## Příklad teplotně závislé konduktivity

Laserové měření tepelné difuzivity metamorfovaných hornin Moldanubika (*Nabelek, 2010*):

$$D = 1.214 \cdot 10^{-6} \times \exp\left(\frac{-T}{285.2}\right) + 3.19 \cdot 10^{-7}$$

Závislost měrné tepelné kapacity na teplotě byla pro tyto horniny definována (*Whittington, 2009*):

$$c = 726 + 1.53 \times T - 1.6 \cdot 10^{-3} \times T^2 + 6.64 \cdot 10^{-7} \times T^3$$

Tyto odvozené veličiny lze využít pro výpočet tepelné konduktivity:

$$\lambda = \rho \cdot c \cdot D$$

# Numerické řešení evoluční rovnice vedení tepla

Aproximaci řešení rovnice (6) získáme

- nahrazením druhé prostorové derivace teploty konečnou diferencí druhé derivace

$$f''(x) \approx \frac{f(x-\Delta x) - 2f(x) + f(x+\Delta x)}{\Delta x^2} \text{ pro všechny } x_i \text{ pro } i = 1 \dots n-2$$

- nahrazením první prostorové derivace teploty a konduktivity centrální konečnou diferencí první derivace

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \text{ pro všechny } x_i \text{ pro } i = 1 \dots n-2$$

- nahrazením první derivace teploty podle času konečnou diferencí první derivace zprava

$$f'(x) \approx \frac{f(x+\Delta x) - f(x)}{\Delta x} \text{ pro všechny } x_i \text{ pro } i = 0 \dots n-1$$

Pro body  $x_1$  a  $x_{n-1}$  použijeme rovnice odvozené z okrajových podmínek.



# Numerické řešení evoluční rovnice vedení tepla II

V bodech  $x_i$  pro  $i = 1 \dots n - 2$  hodnotu hustoty a měrné tepelné kapacity nahradíme průměrnou hodnotou dvou nejbližších hodnot:

$$\bar{\rho}_{x_i} = \frac{\rho_{i-1} + \rho_i}{2}$$

$$\bar{c}_{x_i} = \frac{c_{i-1} + c_i}{2}$$

Pro každý diskrétní bod  $x_i$  pro všechny  $x_i$  pro  $i = 1 \dots n - 2$  pak platí:

$$\begin{aligned} \bar{\rho}_{x_i} \cdot \bar{c}_{x_i} \cdot \frac{T_{x_i}^t - T_{x_i}^{t-\Delta t}}{\Delta t} = & \frac{\lambda_i - \lambda_{i-1}}{\Delta x} \cdot \frac{T_{x_i+\Delta x}^t - T_{x_i-\Delta x}^t}{2\Delta x} \\ & + \bar{\lambda}_{x_i} \cdot \frac{T_{x_i-\Delta x}^t - 2T_{x_i}^t + T_{x_i+\Delta x}^t}{\Delta x^2} + \bar{H}_{x_i} \end{aligned}$$

# Numerické řešení evoluční rovnice vedení tepla III

Úpravou dostaneme

$$-\frac{\lambda_{i-1}}{\Delta x^2} T_{x_{i-1}}^t + \left[ \frac{(\rho_{i-1} + \rho_i) \cdot (c_{i-1} + c_i)}{4\Delta t} + \frac{\lambda_{i-1} + \lambda_i}{\Delta x^2} \right] T_{x_i}^t - \frac{\lambda_i}{\Delta x^2} T_{x_{i+1}}^t = \frac{H_{i-1} + H_i}{2} + \frac{(\rho_{i-1} + \rho_i) \cdot (c_{i-1} + c_i)}{4\Delta t} T_{x_i}^{t-\Delta t}$$

Po substituci

$$\begin{aligned} l_i &= \frac{\lambda_{i-1}}{\Delta x^2} \\ r_i &= \frac{\lambda_i}{\Delta x^2} \\ m_i &= \frac{(\rho_{i-1} + \rho_i) \cdot (c_{i-1} + c_i)}{4\Delta t} \end{aligned}$$

dostaneme

$$-l_i \cdot T_{x_{i-1}}^t + (m_i + l_i + r_i) T_{x_i}^t - r_i \cdot T_{x_{i+1}}^t = \frac{(H_{i-1} + H_i)}{2} + m_i \cdot T_{x_i}^{t-\Delta t}$$

# Numerické řešení - schéma BTCS

V maticovém tvaru lze soustavu rovnic zapsat  $\mathbf{A}x = b$  kde

$$\mathbf{A} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -l_1 & m_1 + l_1 + r_1 & -r_1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -l_{n-2} & m_{n-2} + l_{n-2} + r_{n-2} & -r_{n-2} \\ 0 & 0 & 0 & 0 & -2l_{n-1} & \frac{\rho_{n-2} c_{n-2}}{\Delta t} + 2l_{n-1} \end{vmatrix}$$

$$b = \begin{vmatrix} T_{def} \\ \frac{(H_0 + H_1)}{2} + m_1 \cdot T_{x_1}^{t-\Delta t} \\ \vdots \\ \frac{(H_{n-3} + H_{n-2})}{2} + m_{n-2} \cdot T_{x_{n-2}}^{t-\Delta t} \\ H_{n-2} - \frac{2q}{\Delta x} + \frac{\rho_{n-2} c_{n-2}}{\Delta t} \cdot T_{x_{n-1}}^{t-\Delta t} \end{vmatrix}$$

## Ukázka kódu pro evoluční řešení - BTCS

```
def btcs(t, dt, k, H, dx, rho, c, q=-0.03, T0=0):
    n = len(k)+1
    # substituce
    Hm = (H[:-1] + H[1:])/2
    li, ri = k[:-1]/dx**2, k[1:]/dx**2
    mi = (rho[:-1] + rho[1:])*(c[:-1] + c[1:])/(4*dt)
    # diagonaly
    dl = hstack((-li, -2*k[-1]/dx**2, 0))
    dh = hstack((1, mi+li+ri,
                 rho[-1]*c[-1]/dt + 2*k[-1]/dx**2))
    du = hstack((0, 0, -ri))
    # matice soustavy
    A = spdiags([dl, dh, du], [-1, 0, 1], n, n, 'csr')
    #vektor pravy strany
    b = hstack((T0, Hm + mi*t[1:-1],
                 H[-1]-2*q/dx + t[-1]*rho[-1]*c[-1]/dt))
    return spsolve(A, b)
```