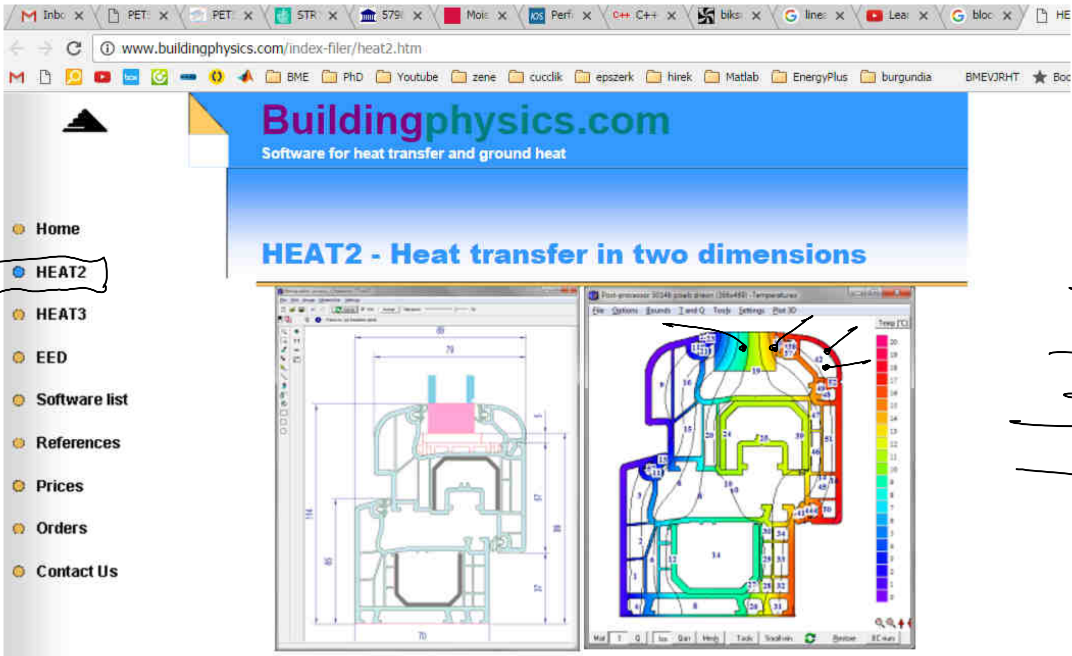
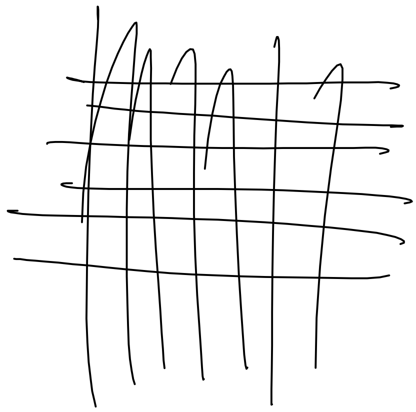
00 - Heat 2D

2016. szeptember 23.
9:12

Heat 2D - free light version

www.buildingphysics.com





Now supporting over 40 languages!

f(x,y)

...

DIN (Deutsches Institut für Normung, DIN V 4108-4) is also available.

· Extensive window frame analysis has been implemented according to ISO 10077.

Support and manuals also available in German, see www.buildingphysics.de

For sales and support in the CIS-countries, Latvia, Lithuania, and Estonia, see www.buildingphysics.ru

Click here for current version update info

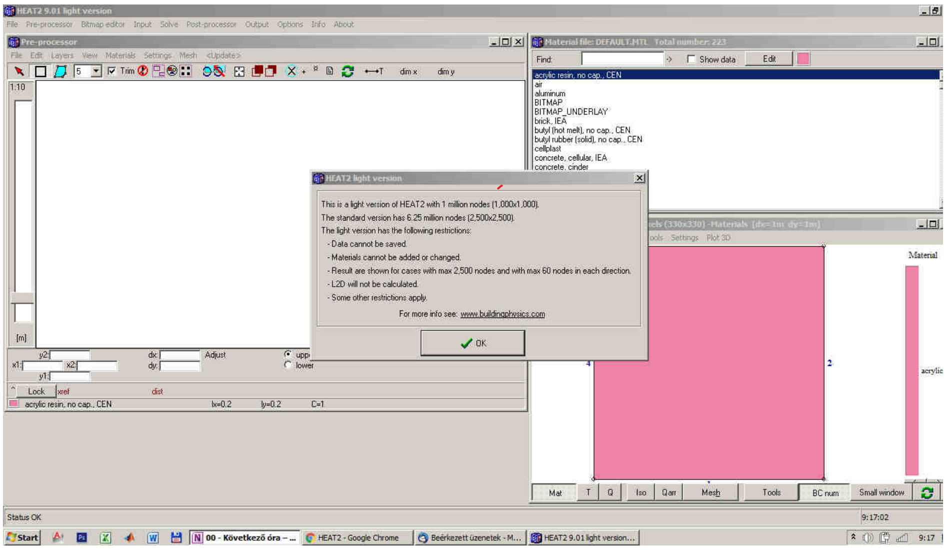
Free light version: Click here to download a free light version

YouTube clips:

· Input example using rectangles, polygons, and bitmaps

· Input example using bitmaps. Also shows a bitmap of a window frame that is imported and filled with different materials.

Tips for reading for beginners: For a quick start read Chapter 4 (pages 23-27) in Manual HEAT2 5.0. The example in chapter 8 (pages 117-121) would also give a short introduction. After this, look at the update manuals for HEAT2 6.0, HEAT2 7.0, etc. Also see the EN ISO test cases: **[ISO 10211 & 10077-2**


HEAT2 9.01 light version
Pre-processor
Material file: DEFAULT.MTL Total number: 223
acrylic resin, no cap., CEN
air
aluminum
BITMAP
BITMAP_UNDERLAY
brick, IEA
butyl (hot melt), no cap., CEN
butyl rubber (solid), no cap., CEN
cellplast
concrete, cellular, IEA
concrete, cinder
HEAT2 light version
This is a light version of HEAT2 with 1 million nodes (1,000x1,000).
The standard version has 6.25 million nodes (2,500x2,500).
The light version has the following restrictions:
- Data cannot be saved.
- Materials cannot be added or changed.
- Result are shown for cases with max 2,500 nodes and with max 60 nodes in each direction.
- L2D will not be calculated.
- Some other restrictions apply.
For more info see: www.buildingphysics.com
OK
Material
acrylic resin, no cap., CEN
Status OK
9:17:02
Start
HEAT2 - Google Chrome
HEAT2 9.01 light version...

LBNL Therm

https://windows.lbl.gov/software/therm/therm.html

https://windows.lbl.gov/software/therm/therm.html

BME PhD Youtube zene cucclik epszerk hirek Matlab EnergyPlus burgundia BMEVJRHT Bookmarks Munkak space

THERM

THERM 6.3 For NFRC Certification and modeling complex glazing systems	THERM 7.4 For modeling vacuum glazing, deflected glass, vertical venetian blinds, cellular shades and perforated screens
Forum For questions about THERM	Forum For questions about THERM
Knowledge Base (Check here first if you are experiencing a problem with the software)	Knowledge Base (Check here first if you are experiencing a problem with the software)
Documentation	Documentation
Tutorials	Tutorials

Two-Dimensional Building Heat-Transfer Modeling

THERM is a state-of-the-art computer program developed at Lawrence Berkeley National Laboratory (LBNL) for use by building component manufacturers, engineers, educators, students, architects, and others interested in heat transfer. Using THERM, you can model two-dimensional heat-transfer effects in building components such as windows, walls, foundations, roofs, and doors; appliances; and other products where thermal bridges are of concern. THERM's heat-transfer analysis allows you to evaluate a product's energy efficiency and local temperature patterns, which may relate directly to problems with condensation, moisture damage, and structural integrity.

THERM's two-dimensional conduction heat-transfer analysis is based on the finite-element method, which can model the complicated geometries of building products. See Components for more details.

THERM can be used with the Berkeley Lab WINDOW program. THERM's results can be used with WINDOW's center-of-glass optical and thermal models to determine total window product U-factors and Solar Heat Gain Coefficients. These values can be used, in turn, with the RESFEN program, which calculates total annual energy requirements in typical residences throughout the United States.

Components

...

THERM 7.4

Last Updated: 10/03/2015

If you find bugs, or have comments about this version, we now have an online forum where you can ask questions and respond to questions by others. Getting feedback from users is how we improve the program.

THERM 7 contains many new modeling features, including:

- Deflection Model
- Vacuum Glazing
- Vertical Louvered Blinds
- Perforated Screens
- Honeycomb shades
- Dynamic Glazing (Thermochromic and Electrochromic)

Latest Version

THERM 7.4.3
(7.4.3)
(10/03/2015)

Release Notes -- *Please read these before running this version !*

This version is compatible with WINDOW 7.4.6.

- **If you try to import THERM 7.4.3 files into earlier versions of WINDOW 7.3, WINDOW may crash; in this case upgrade to this latest version of WINDOW 7.4**

...

Windows
& daylighting

Download Registration

Before you can download software from this website you need to login. If you don't have an account, you need to Create an account first.

This registration page will not work unless you have enabled Cookies in your browser.

Sign in with your Account

E-mail: bakonyi.dani@gmail.com

Password: ••••••••••••

You need to login first.
Please enter your login details.

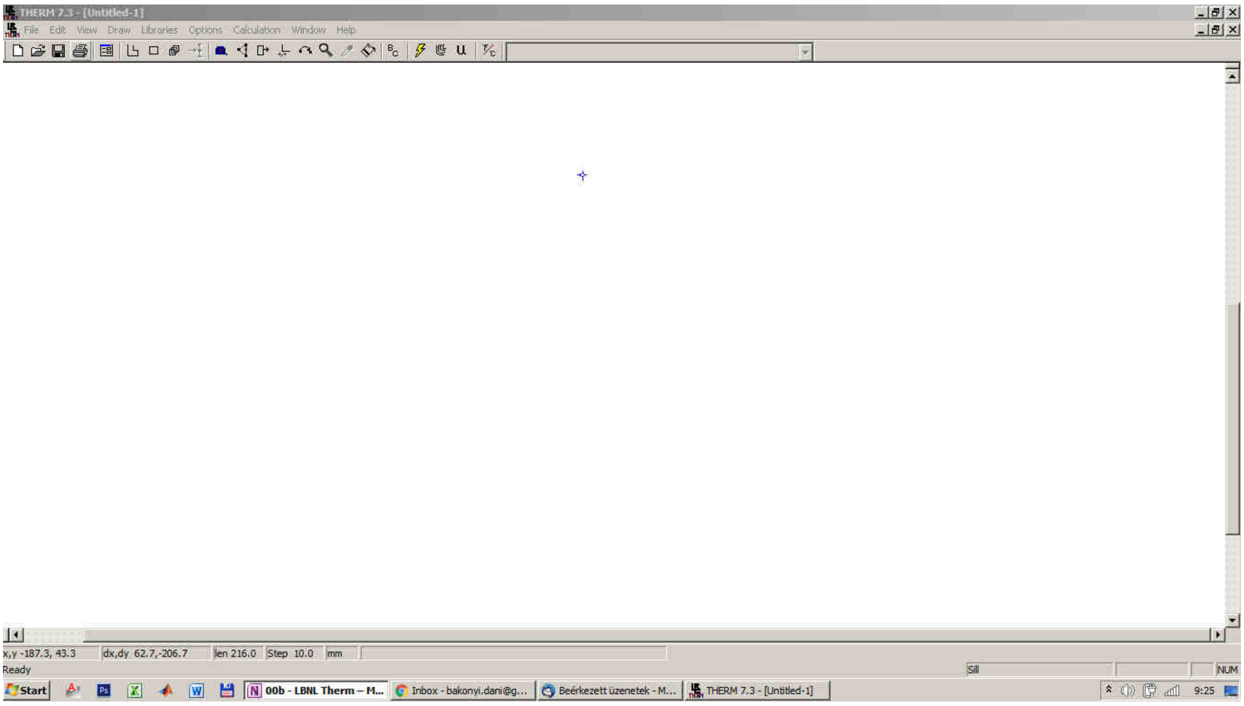
Remember me on this computer.

Login

Send my password

Don't have an Account yet?
Create an account now

If you have questions or problems with the registration, please contact WindowHelp@lbl.gov



01 - lineáris algebra

2016. szeptember 23.
11:52

vektor $\bar{v}$ v. $\vec{v} = (v_1, v_2, v_3, \dots)$

mátrix $A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$

3×2

oszlopok száma

sorok száma

vektor műveletek:

$\vec{u} + \vec{v} = (u_1 + v_1, u_2 + v_2, u_3 + v_3)$

$\vec{u} - \vec{v} = (u_1 - v_1, u_2 - v_2, u_3 - v_3)$

$\alpha \cdot \vec{u} = (\alpha u_1, \alpha u_2, \alpha u_3)$ skalárral szorzás

$\|\vec{u}\| = \sqrt{u_1^2 + u_2^2 + u_3^2}$ norm

$\vec{u} \cdot \vec{v} = u_1 v_1 + u_2 \cdot v_2 + u_3 \cdot v_3$ skaláris szorzat

$\|\vec{u}\| \cdot \|\vec{v}\| \cdot \cos \varphi$

$\vec{u} \times \vec{v} = (u_2 v_3 - u_3 v_2, u_3 v_1 - u_1 v_3, u_1 v_2 - u_2 v_1)$ vektoriális szorzat

$\|\vec{u} \times \vec{v}\| = \|\vec{u}\| \|\vec{v}\| \sin \varphi$

$\vec{u} \times \vec{v} \neq \vec{v} \times \vec{u}$

Mátrix műveletek:

$A + B = C$ $c_{ij} = a_{ij} + b_{ij}$

$A - B = C$ $c_{ij} = a_{ij} - b_{ij}$

$A \cdot B = C$ $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

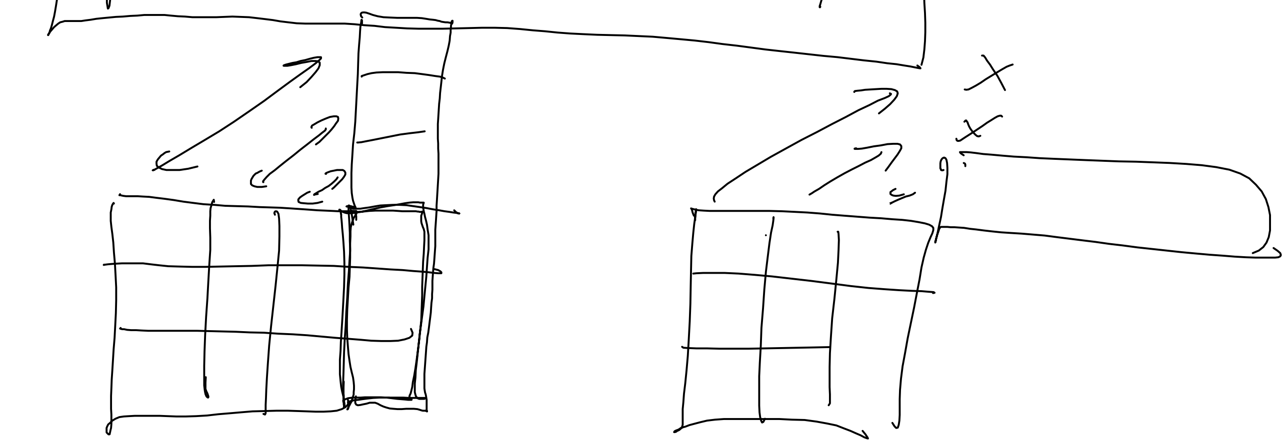
$B \rightarrow \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{vmatrix}$

A n

$$
A \quad
\begin{array}{c}
\\ m
\end{array}
\begin{array}{c}
n \\
\left|\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}\right|
\end{array}
\quad
\begin{array}{c}
n\left|\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}\right| \\
\left|\begin{matrix} 30 & & \\ & & \\ & & \end{matrix}\right| \; C
\end{array}
$$

1. A oszlopai száma = B sorai száma
2. C – egy sor alatt A
   – egy oszlop alatt B

$$
\left|\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}\right| \left|\begin{matrix} 1 \\ 1 \\ 1 \end{matrix}\right| = \left|\begin{matrix} 6 \\ 15 \\ \vdots \end{matrix}\right|
$$



A $\qquad$ A^{-1} ← inverz

$A^{-1} \cdot A = I$ ← egység mátrix
ididentity-matrix

együtthatók

$A \cdot \cancel{T}X = b$ ← ismert elemek vektora

↑ ismeretlenek vektora

ismeretlenek vektora

$$\underbrace{A^{-1} A}\cdot T = A^{-1}\cdot b$$

$$I \cdot T = A^{-1} b$$

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} T_1 \\ T_2 \\ T_3 \end{vmatrix} = \begin{vmatrix} T_1 \\ T_2 \\ T_3 \end{vmatrix}$$

$$T = A^{-1}\cdot b$$

---

elimináció   Gauss–Jordan elimináció

$$1x_1 + 2x_2 = 5$$
$$3x_1 + 9x_2 = 21$$

$$\begin{vmatrix} \boxed{1} & 2 \\ 3 & 9 \end{vmatrix} \begin{vmatrix} x_1 \\ x_1 \end{vmatrix} = \begin{vmatrix} 5 \\ 21 \end{vmatrix}$$

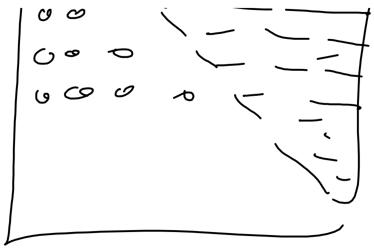
- kicserélni sorokat
- egyik sor nX hozzáadom egy másik sorhoz
- egyik sor szorzása egy konstanssal

$n = 16382$

n

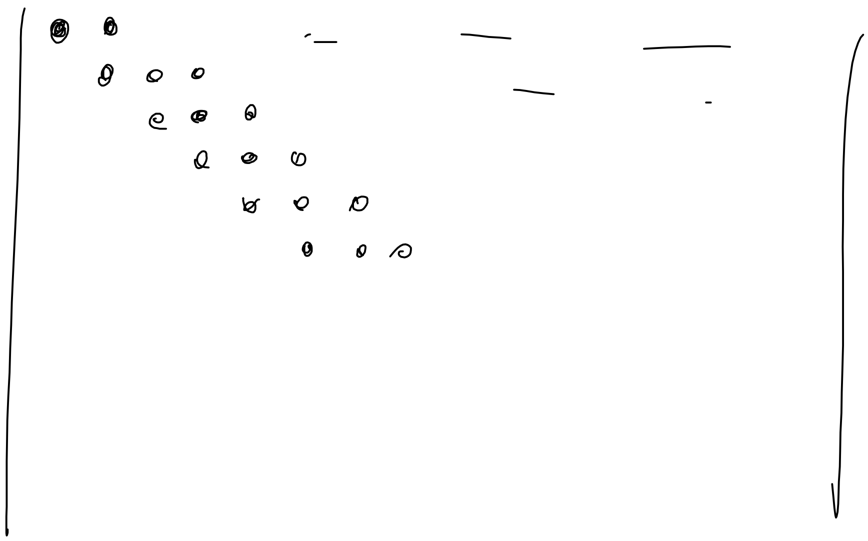
$$\begin{vmatrix} 1 & 2 \end{vmatrix} \begin{vmatrix} x_1 \\ \; \end{vmatrix} = \begin{vmatrix} 5 \\ \; \end{vmatrix}$$

$$\begin{vmatrix} 1 & 2 \\ 0 & 3 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} = \begin{vmatrix} 5 \\ 6 \end{vmatrix} \longrightarrow x_2$$

u 

$$\begin{vmatrix} 1 & 2 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} = \begin{vmatrix} 5 \\ 2 \end{vmatrix} \qquad / :3$$

$$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} = \begin{vmatrix} 1 \\ 2 \end{vmatrix} \qquad / -2 \times 2.\text{sor}$$



ritka
mátrix

$$x_1 + 2x_2 = 1$$

$$2x_1 + 4x_2 = 2$$

$$\begin{vmatrix} 1 & 2 \\ \cancel{2} & 4 \end{vmatrix}$$

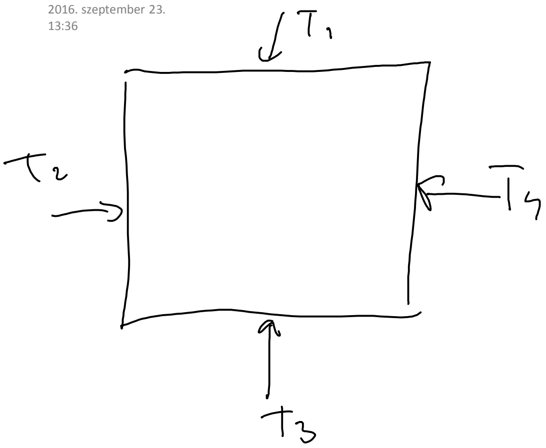
$$\begin{vmatrix} 1 & 2 \\ 0 & 0 \end{vmatrix}$$

$$\begin{vmatrix} 4 \\ 5 \\ 0 \end{vmatrix}$$

$$\begin{vmatrix} 1 & 2 & 3 \end{vmatrix}$$

04 - programozás példa

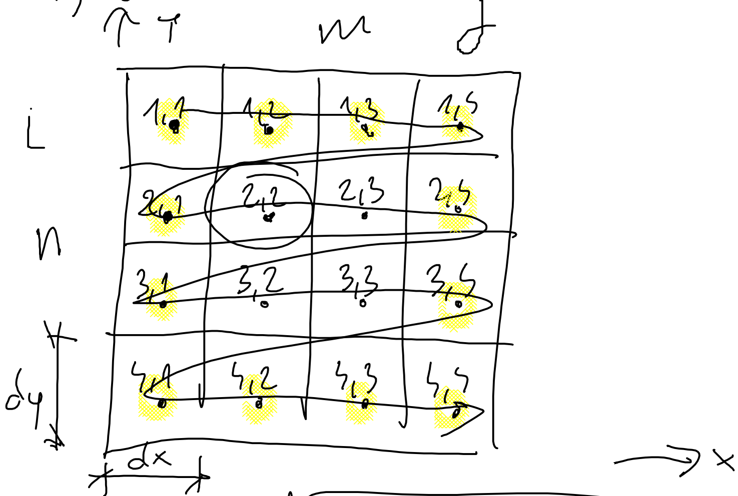
2016. szeptember 23.
13:36



$$\lambda_x \frac{\partial^2 T}{\partial x^2} + \lambda_y \frac{\partial^2 T}{\partial y^2} = 0$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x+dx) - 2T(x) + T(x-dx)}{dx^2}$$

1.) diszkretizáció



$i = 1 \quad j = 1$

$(i-1) * nx + j$

1,1 : $\boxed{T_{1,1} = 10}$

...

2,1 $\quad T_{2,1} = 10$

2,2

$$+\frac{\lambda_x}{dx^2} T_{2,1} - \frac{2\lambda_x}{dx^2} T_{2,2} + \frac{\lambda_x}{dx^2} T_{2,3} + \frac{\lambda_y}{dy^2} T_{1,2} - \frac{2\lambda_y}{dy^2} T_{2,2} + \frac{\lambda_y}{dy^2} T_{3,2} = 0$$

2.) vektorizáció

$K \cdot T = b$

T mátrix

$$T = \begin{bmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \\ T_{1,4} \\ \vdots \end{bmatrix}$$

$$T = \begin{bmatrix} T_{1,3} \\ T_{1,4} \\ T_{2,1} \\ T_{2,2} \\ T_{2,3} \\ T_{3,4} \\ T_{3,1} \\ \vdots \end{bmatrix}$$

K mátrix

$$\begin{array}{ccccccccc} 1,1 & 1,2 & 1,3 & 1,4 & 2,1 & 2,2 & 2,3 & \dots & 3,2 \\ \boxed{1} & 0 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & 0 & \dots & & & \\ 0 & 0 & 1 & 0 & 0 & \dots & & & \\ 0 & 0 & 0 & 1 & 0 & \dots & & & \\ 0 & 0 & 0 & 0 & 1 & \dots & & & \\ & \frac{\lambda y}{dy^2} & 0 \dots 0 & & \frac{\lambda x}{dx^2} & -\frac{2\lambda x}{dx^2} - \frac{2\lambda y}{dy^2} & \frac{\lambda x}{dx^2} & \dots & \frac{\lambda y}{dy^2} \end{array} \begin{bmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \\ T_{1,4} \\ T_{2,1} \\ T_{2,2} \\ T_{2,3} \\ \vdots \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 0 \end{bmatrix}$$