

CNN, 7. gyakorlat

2017. tavasz

Mai feladatok

1. hőterjedés vizsgálata: 1D, 2D, melegítés
2. folyadék betöltése csatornarendszerbe
3. utazóhullám 1D-ben

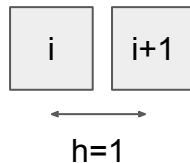
Ismétlés előadásról

Hőterjedés egyenlete:
$$\frac{\partial u}{\partial t} - \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) = 0$$

- u : hőmérséklet érték
- α : skálázási faktor

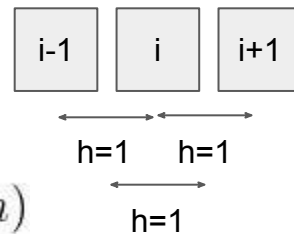
Térbeli első derivált közelítése CNN-en (**1D**):

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$



Térbeli második derivált közelítése CNN-en (**1D**):

$$f''(x) = \lim_{h \rightarrow 0} \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h} = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



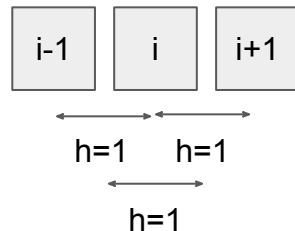
ez visszacsatoló template-ként: $A = [1, -2, 1]$

Ismétlés előadásról

Térbeli második derivált közelítése CNN-en (**1D**):

$$f''(x) = \lim_{h \rightarrow 0} \frac{\frac{f(x+h)-f(x)}{h} - \frac{f(x)-f(x-h)}{h}}{h} = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

ez visszacsatoló template-ként: $\mathbf{A} = [1, -2, 1]$



Kiterjesztés **2D**-re (rajz és formalizmus nélkül):

$$\mathbf{A} = \begin{bmatrix} 0.5 & 1 & 0.5 \\ 1 & -6 & 1 \\ 0.5 & 1 & 0.5 \end{bmatrix}$$

Hőterjedés vizsgálata, 1D

Egyenetlenül felmelegített fémrúdban hőkiegyenlítődés. A megvalósítás során egy 3-oszlopos mátrixot fogunk használni (kompatibilitási okokból).

Lépések:

- a `temlib_diff.m` függvénykönyvtárat egészítsd ki egy **HEAT** template-tel:
 - **A** template: középső oszlopban az 1D-s, második deriváltas értékek; **B**, **z** értéke nullamátrix és nulla
- készíts egy `heat1D.m` szkript-et:
 - matCNN inicializálása, függvénykönyvtár megadása
 - kezdeti állapot: $n \times 3$ -as mátrix, soronként az alábbi értékekkel: $2e^{0.2(-idx+0.5)} - 1$
(ahol ' idx ' az adott sor indexe)
 - bemenet: csupa nulla mátrix
 - határfeltétel: *zero-flux*
 - időlépés: *0.1*
 - **HEAT** template betöltése
 - érdemes egy `for`-ciklusban mindig csak 1 iterációt futtatni, és a ciklusmagon belül kirajzolni a szimulátor kimeneti termináljáról letölthető tömb 2. oszlopát, hagyományos `plot` utasítással
 - azonnali rajzolás kikényszerítése: `drawnow`
 - futtassunk 100 ciklus-iterációt

Hőterjedés vizsgálata, 2D

Adott egy 2D lemez, aminek egy része (egy minta mentén) fel van melegítve. Az idő előrehaladtával kiegyenlítődik a hőmérséklete.

Lépések:

- a `temlib_diff.m` függvénykönyvtárat egészítsd ki egy **HEAT2D** template-tel:
 - **A** template: mint ismételtnél néztük (1D template minden irányba, kumulálva); **B**, **z** nulla
- készíts egy **heat2D.m** szkript-et:
 - matCNN inicializálása, függvénykönyvtár megadása
 - kezdeti állapot: az `init_heat.bmp` tartalmazza, betölthetjük a kiadott `loadCNN.m` függvénnyel
 - bemenet: csupa nulla mátrix
 - határfeltétel: *zero-flux*
 - időlépés: *0.01*
 - **HEAT2D** template betöltése
 - érdemes egy `for`-ciklusban mindig csak 3 iterációt futtatni, és a ciklusmagon belül kirajzolni a szimulátor kimeneti termináljáról letölthető tömböt, hagyományos `surf` utasítással
 - azonnali rajzolás kikényszerítése: `drawnow`
 - z-tengely mentén a tartomány rögzítése: `set(gca, 'ZLim', [-1, 1]);`
 - futtassunk 100 ciklus-iterációt

Hőterjedés vizsgálata, folyamatos melegítéssel

Adott egy kezdetben egyen-hőmérsékletű 2D lemez, aminek egy részét (egy minta mentén) elkezdjük melegíteni. Az idő előrehaladtával a melegítés helyén nő a hőmérséklete, az azon kívüli régiókon kihűlés következik be.

Lépések:

- a **temlib_diff.m** függvénykönyvtárat egészítsd ki egy **HEATING** template-tel:
 - **A** template: megegyezik a **HEAT2D** template visszacsatolásával, de érdemes az értékeket egy nagyságrenddel felskálázni
 - **B**: csak a centrális cella kapja meg, csak a ráeső bemeneti pixelt, leskálázva 1 nagyságrenddel
 - **z**: nulla

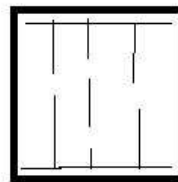
Hőterjedés vizsgálata, folyamatos melegítéssel

folytatás:

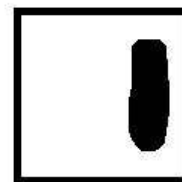
- készíts egy `heating.m` szkript-et:
 - `matCNN` inicializálása, függvénykönyvtár megadása
 - bemenet: az `init_heat.bmp` tartalmazza, betölthetjük a kiadott `loadCNN.m` függvénnyel
 - kezdeti állapot: csupa nulla mátrix
 - határfeltétel: *zero-flux*
 - időlépés: *0.001*
 - **HEATING** template betöltése
 - érdemes egy `for`-ciklusban mindig csak 1 iterációt futtatni, és a ciklusmagon belül kirajzolni a szimulátor kimeneti termináljáról letölthető tömböt, hagyományos `surf` utasítással
 - azonnali rajzolás kikényszerítése: `drawnow`
 - z-tengely mentén a tartomány rögzítése: `set(gca, 'ZLim', [-0.01, 0.01]);`
 - futtassunk 100 ciklus-iterációt

Folyadékterjedés vizsgálata

Egy adott alakú csatornarendszert fogunk feltölteni, egy bemeneti résen keresztül.



csatorna
fala



betöltési rés

Lépések:

- a `temlib_diff.m` függvénykönyvtárat egészítsd ki egy **FLOW** template-tel:

$$\begin{array}{rcccl} & 0 & 1 & 0 & & 0 & 0 & 0 \\ \mathbf{A} = & 1 & -2.9 & 1 & \mathbf{B} = & 0 & 1 & 0 & \mathbf{z} = & 1 \\ & 0 & 1 & 0 & & 0 & 0 & 0 \end{array}$$

Folyadékterjedés vizsgálata

folytatás:

- készíts egy `flow.m` szkript-et:
 - `matCNN` inicializálása, függvénykönyvtár megadása
 - bemenet: a `flow_input.bmp` tartalmazza, betölthetjük a kiadott `loadCNN.m` függvénnyel
 - kezdeti állapot: csupa nulla mátrix
 - használni fogjuk a *maszkolás* lehetőségét: ez lehetővé teszi, hogy egy adott template a cellatömb csak egy adott részén működjön (létezik bizonyos hardveres megvalósításban is):
 - `mCNN.UseMask = 1 ; % ezzel jelezzük, hogy maszkolást szeretnénk`
 - `mCNN.MASK = loadCNN('gates.bmp') ; % ez maga a maszk; (értelemszerűen az inverze a csatorna határait ábrázoló képnek)`
 - határfeltétel: *zero-flux*
 - időlépés: *0.1*
 - **FLOW** template betöltése
 - érdemes egy `for`-ciklusban mindig csak 5 iterációt futtatni, és a ciklusmagon belül kirajzolni a szimulátor kimeneti termináljáról letölthető tömböt, hagyományos `surf` utasítással
 - azonnali rajzolás kikényszerítése: `drawnow`
 - futtassunk 500 ciklus-iterációt

Utazóhullám vizsgálata, 1D

Kezdeti egységnyi “energiacsomagból” kialakuló utazóhullám (nem konstans amplitúdóval). A megvalósítás során egy 3-oszlopos mátrixot fogunk használni (kompatibilitási okokból).

Lépések:

- a **temlib_diff.m** függvénykönyvtárat egészítsd ki egy **WAVE** template-tel:
 - **A** template: középső oszlop felső sorában egy 1-es, egyébként 0 (1D-ben “a felső szomszédot másolja lefelé”);
 - **B**, **z** értéke nullamátrix és nulla

Utazóhullám vizsgálata, 1D

folytatás:

- készíts egy **wave.m** szkript-et:
 - matCNN inicializálása, függvénykönyvtár megadása
 - kezdeti állapot: $nx3$ -as mátrix: az első öt sor értéke +1, a többié -1
 - bemenet: csupa nulla mátrix
 - határfeltétel: *periodikus*
 - időlépés: *0.1*
 - **WAVE** template betöltése
 - érdemes egy **for**-ciklusban mindig csak 5 iterációt futtatni, és a ciklusmagon belül kirajzolni a szimulátor kimeneti termináljáról letölthető tömb 2. oszlopát, hagyományos **plot** utasítással
 - azonnali rajzolás kikényszerítése: **drawnow**
 - kirajzolás dinamikatarományának rögzítése: **set(gca, 'YLim', [-1.2 1.2]);**
 - futtassunk 100 ciklus-iterációt