Numerické simulace, optimalizace, AI a HPC

Tomáš Oberhuber, M. Beneš, R. Fučík, J. Mikyška, P. Strachota

Katedra matematiky,
Fakulta jaderná a fyzikálně inženýrská,
České vysoké učení technické v Praze

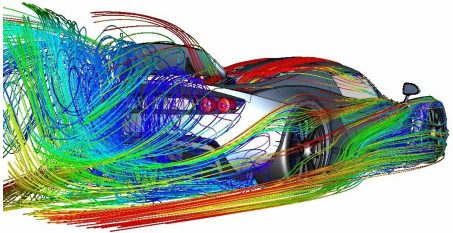
Seminář současné matematiky

CFD - výpočetní dynamika tekutin

Navierovy-Stokesovy rovnice

$$\vec{u}_t + \vec{u} \cdot \nabla \vec{u} + \nabla p = \nu \Delta \vec{u} \text{ na } (0,T) \times \Omega,$$
$$\nabla \cdot \vec{u} = 0 \text{ na } (0,T) \times \Omega,$$

+ příslušné okrajové a počáteční podmínky



Millennium Prize Problems:

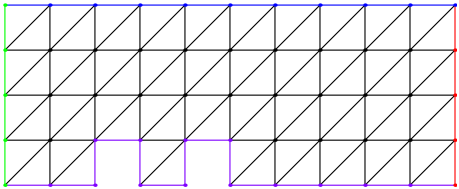
1. Problém P versus NP
2. Hodgeova domněnka
3. Poincarého domněnka (vyřešena)
4. Riemannova hypotéza
5. Yangova-Millsova teorie a hypotéza hmotnostních rozdílů
6. **Navierovy-Stokesovy rovnice**
7. Birchova a Swinnerton-Dyerova domněnka

Za vyřešení každého z nich je vypsána odměna 1,000,000 $.

Navierovy-Stokesovy rovnice pro nestlačitelné proudění

Diskretizace prostorových derivací:

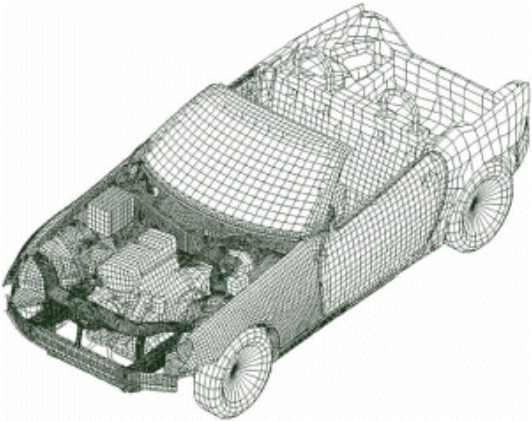
- *metoda konečných diferencí*
  - pouze ortogonální sítě
- *metoda konečných objemů*
- *metoda konečných prvků*



Diskretizace časových derivací:

- pomocí časového kroku $\tau = T/n$
- přechod z času $k\tau \to (k+1)\tau$
- *explicitní metoda*
  - přímý výpočet
  - je nutný **malý časový krok**
- *semi-implicitní metoda*
  - řešením soustavy lineárních rovnic
  - velikost lineární soustavy je úměrná počtu buněk sítě
  - soustava je řídká
  - lze volit **delší časový krok**
- *plně implicitní metoda*
  - Newtonova metoda

Navierovy-Stokesovy rovnice pro nestlačitelné proudění



Numerická aproximace

- je potřeba efektivně řešit ...
  - velké množství
  - velkých lineární soustav
  - s řídkými maticemi
- iterativní metody Krylovových podprostorů
  - CG, BiCGStab, GMRES
- efektivní předpodmínění
  - ILU, SPAI, multigridní metody
- **jde o výpočetně velmi náročnou úlohu**

Numerické výpočty vyžadují stále větší výpočetní výkon
běžné procesory nejsou schopny těmto požadavkům vyhovět

CPU - Central Processing Units

Velká a sofistikovaná výpočetní jádra

- lepší zpracování obecného kódu
- lepší efektivita při náhodných přístupech do paměti

GPU - Graphics Processing Units

Velký počet jednoduchých výpočetních jednotek

- velký výpočetní výkon
- optimalizováno pro čtení velkých bloků dat

Reportovaná urychlení dosažitelná pomocí GPU

- Výpočetní dynamika tekutin - 50x - 100x urychlení
- Lineární algebra - 10x urychlení
- Strojové učení - 50x urychlení
- Monte Carlo - 10x urychlení

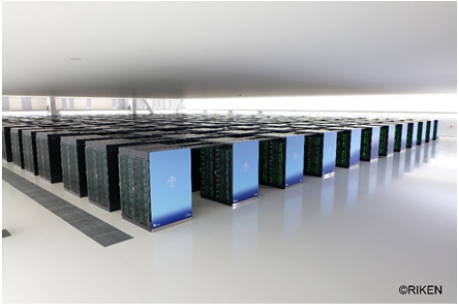
Porovnání GPU a CPU

	Nvidia A100	AMD Instinct MI100	AMD EPYC 2 Rome 7H12
Počet jader	6912 @ 1.41GHz	7680 @ 1.2GHz	64 @ 2.4GHz
Max. výkon	19.5/9.7 TFlops	23/11 TFlops	2 / 1 TFlops
Max. výkon s tenzory	156 /– TFlops	–/–	–/–
Max. RAM	80 GB	32 GB	2 TB
Datová propustnost	2039 GB/s	1200 GB/s	204 GB/s
Energetická náročnost	400 W	300 W	280 W

Superpočítače

www.top500.org

- **Fugaku** - Japonsko
  - 158,976 výpočetních uzlů = **0.5 EFLOPS**
  - 7,630848 procesorových jader
  - paměť 5 PB
  - příkon 30MW
- **Summit** - USA
  - 9,216 výpočetních uzlů = **0.2 EFLOPS**
  - 2,414,592 procesorových jader
  - 27,648 GPU Nvidia Tesla V100
  - paměť 5 PB
  - příkon 10 MW



Komplikace s využitím GPU

Řada numerických algoritmů je výrazně sekvenčních

- je potřeba hledat nové způsoby paralelizace
- vyvíjet zcela nové metody

Programování pro GPU je příliš komplikované pro běžné uživatele

- je potřeba detailní znalost architektury GPU
- vývoj kódu může trvat několikanásobně déle než na CPU
  - skalární součin dvou vektorů $\approx$ 150 řádků kódu

Template Numerical Library

TNL = Template Numerical Library - Tomáš Oberhuber, Jakub Klinkovský

- numerická knihovna vyvíjena na KM, FJFI
- napsána v C++ s využitím nejnovějších vlastností tohoto jazyka
- nabízí unifikované rozhraní pro vícejádrová CPU a GPU
- **www.tnl-project.org**
- MIT licence
- afiliovaný projekt organizace Numfocus (www.numfocus.org)

NUMFOCUS [AFFILIATED PROJECT]

TEMPLATE NUMERICAL LIBRARY

Hlavní vlastnosti knihovny TNL

Šablonové metaprogramování

$$\vec{x} = \vec{a} + 2\vec{b} + 3\vec{c}$$

Běžně lze psát pomocí knihovny Cublas:

```
cublasHandle_t c_h;
cublasSaxpy(c_h,N,1.0,a,1,x,1);
cublasSaxpy(c_h,N,2.0,b,1,x,1);
cublasSaxpy(c_h,N,3.0,c,1,x,1);
```

S pomocí **výrazových šablon** v TNL:

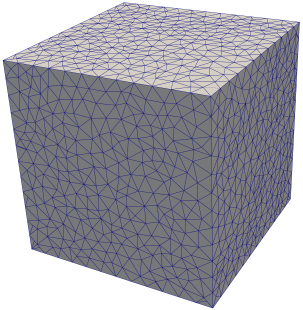
```
x = a + 2 * b + 3 * c;
```

- flexibilní **paralelní redukce** s pomocí **lambda funkcí**
- jednotné rozhraní pro prácí s **hustými a řídkými maticemi**
- iterativní řešiče pro **soustavy lineárních rovnic**
- obecná datová struktura pro **nestrukturované numerické sítě**

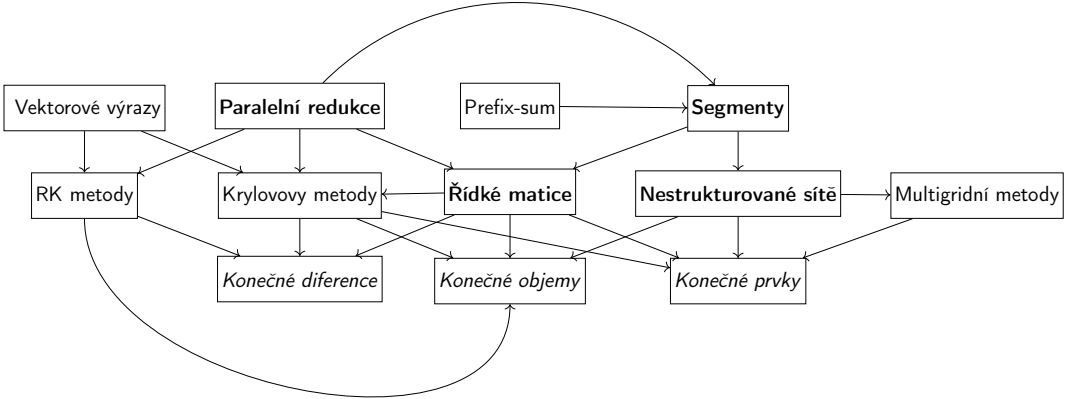
Hlavní vlastnosti knihovny TNL

Obecná datová struktura pro nestrukturované numerické sítě

- základ pro **metody konečných prvků a konečných objemů**
- paralelní řešič pro vícefázové proudění v porézním prostředí (MHFEM)
  [Fučík, Klinkovský, Solovský, Oberhuber, Mikyška, 2019]



Řešení parciálních diferenciálních rovnic

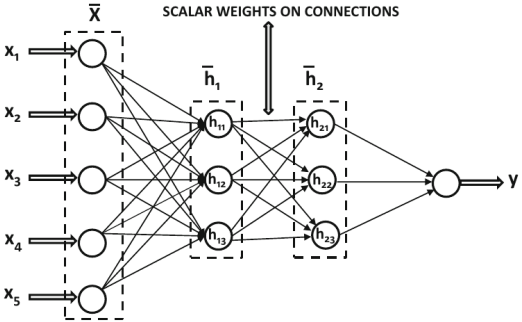


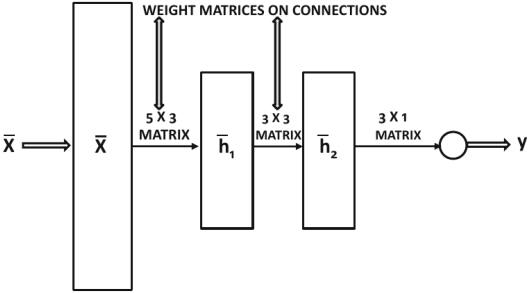
Urychlení dosažená pomocí GPU

Úloha	CPU (OpenMP)	Jádra	GPU (CUDA)	Urychlení
Nestlačitelné NS rce.	Intel i7-3770	4	Tesla K40	**11x**
Vícefázové proudění	Intel i7-5820K	6	Tesla K40	**7.5x**
Segmentace dat z MR	AMD Opt. 6172	12	GeForce GTX 480	**10x**
Anizotropní Willmorův tok	Intel Core 2	4	GeForce GTX 280	**6x**

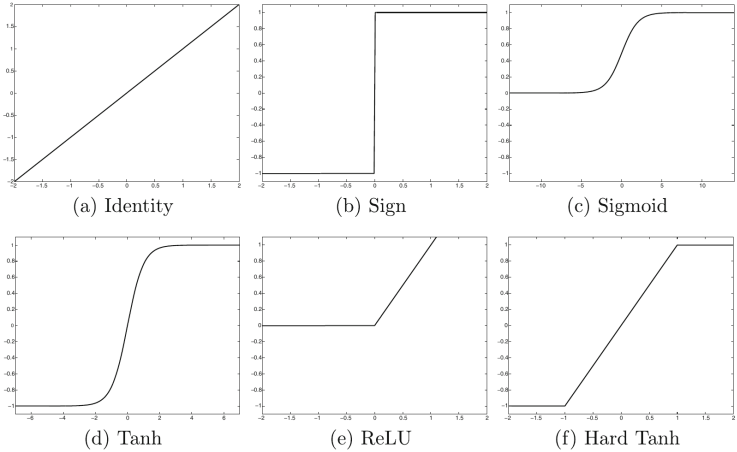
Momentálně nejvíce chybí efektivní předpodmínění pro GPU.

Strojové učení a neuronové sítě





Strojové učení a neuronové sítě



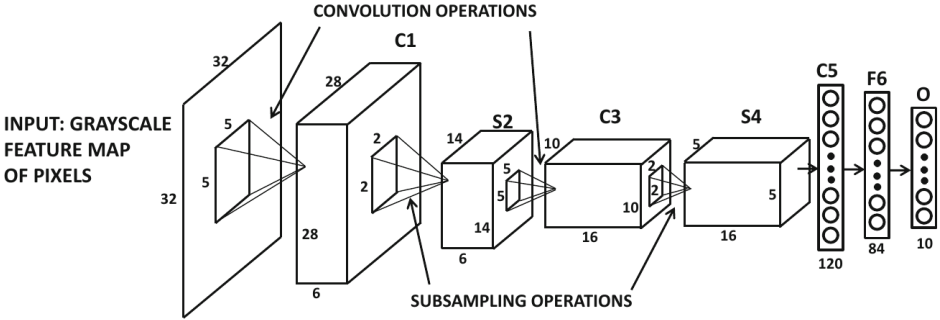
(a) Identity (b) Sign (c) Sigmoid

(d) Tanh (e) ReLU (f) Hard Tanh

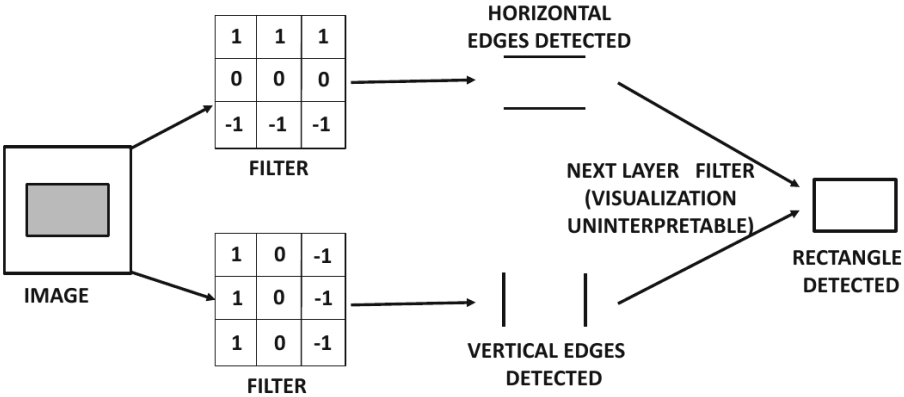
Strojové učení a neuronové sítě

- funkce neuronové síťe je dána nastavením vah u jednotlivých spojů
- to děláme pomocí optimalizace ztrátové funkce $L(\vec{w}, \vec{d})$
- ztrátová funkce na základě známých vstupů a výstupů ohodnocuje, jak dobře neuronová síť reaguje
- optimalizační úlohu řešíme pomocí metody nějvětšího spádu (gradient descent)
- parciální derivace $\frac{\partial L}{\partial w_i}$ napočítáváme pomocí algoritmu **backpropagation**

Konvoluční neuronové sítě a zpracování obrazu



Konvoluční neuronové sítě a zpracování obrazu



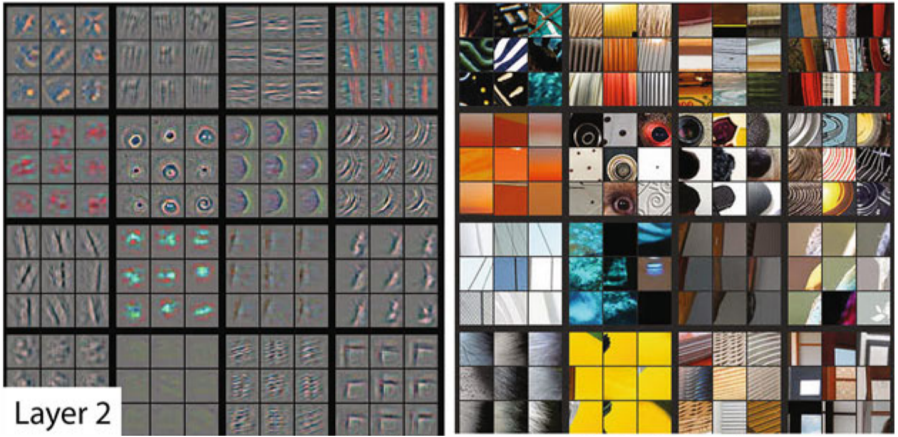
Konvoluční neuronové sítě a zpracování obrazu

ImageNet Large Scale Visual Recognition Challenge (ILSVRC):

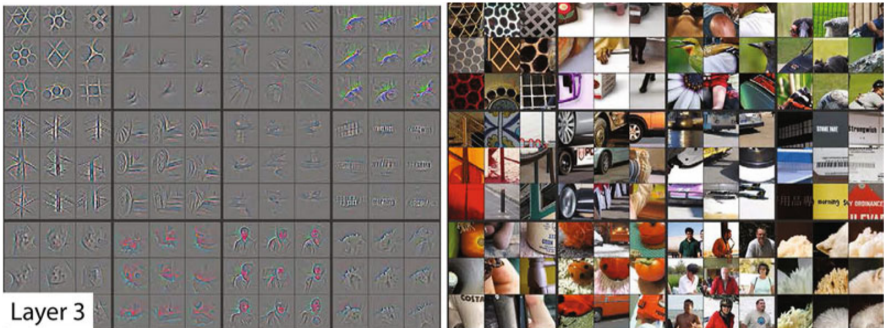
- soutěž algoritmů v rozpoznávání obrazu v letech 2010-2017
- do roku 2012 byla chybovost nejlépe **25%**
- 2012 - AlexNet - **15.4%**
- 2013 - ZFNet - **11.1%**
- 2014 - GoogLeNeT - **6.7%**
- 2015 - ResNet - **3.6%** - srovnatelné s člověkem

Konvoluční neuronové sítě a zpracování obrazu

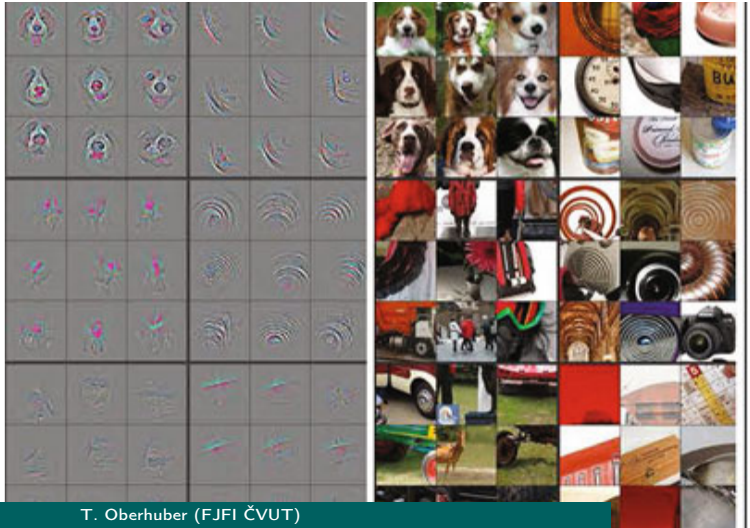
Konvoluční neuronové sítě a zpracování obrazu



Konvoluční neuronové sítě a zpracování obrazu



Konvoluční neuronové sítě a zpracování obrazu

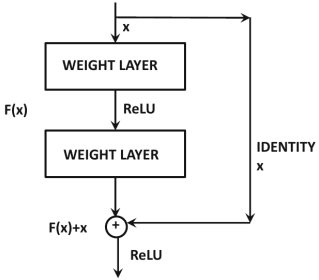


Konvoluční neuronové sítě a zpracování obrazu

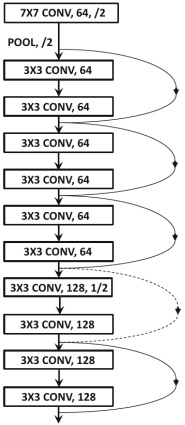
Úspěch neuronových a zejména konvolučních síti byl možný díky:

- výzkumu v oblasti NN - nové optimalizační metody
- dostupnosti velkého množství trénovacích dat
- dostatečnému výpočetnímu výkonu, zejména díky GPU

Reziduální neuronové sítě



(a) Skip-connections in residual module



(b) Partial architecture of *ResNet*

Reziduální neuronové sítě

- aktivace jednotlivých vrstev pak lze popsat jako

$$\vec{x}^{(l+1)} = \vec{x}^{(l)} + F(\vec{x}^{(l)}, \vec{w})$$

- to lze přepsat do tvaru

$$\vec{x}^{(l+1)} = \vec{x}^{(l)} + \Delta t \cdot F(\vec{x}^{(l)}, \vec{w}),$$

  pro $\Delta t = 1$
- tento tvar odpovídá Eulerově metodě pro řešení ODR

$$\frac{\mathrm{d}\vec{x}(\vec{w})}{\mathrm{d}t} = F(\vec{x}, t, \vec{w})$$

E. Weinan, A proposal on machine learning via dynamical systems, Communications in Mathematics and Statistics, vol. 5, no. 1, pp. 1–11, 2017

T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in Advances in Neural Information Processing Systems, pp. 6571–6583, 2018

Adjoint ODR

- můžeme tedy řešit následující úlohu

$$\min_{\vec{w}} L(\vec{x}(\vec{w}), \vec{d}),$$

kde

$$\frac{\mathrm{d}\vec{x}(\vec{w})}{\mathrm{d}t} = F(\vec{x}, t, \vec{w})$$

- úlohu můžeme opět řešit pomocí metody největšího spádu - GD, gradient descent
- k tomu potřebujeme umět napočítat $\frac{\partial L}{\partial w_i}$
- za tím účelem lze odvodit tzv. adjungovanou rovnici (**adjoint equation**)
- ta je vlastně obdobou algoritmu **backpropagation**

N. Heim, T. Pevný, V. Šmídl, Neural Power Units, Advances in Neural Information Processing Systems 33 (2020): 6573-6583

Adjoint PDR

Řešení adjungovaných diferenciálních rovnic

$$
\begin{aligned}
\partial_t \varphi &= \mathcal{D}(\vec{\theta}_1) + F(\vec{\theta}_2) \text{ on } \Omega(\vec{\theta}_3) \times (0, T]\,, \\
\varphi\,|_{t=0} &= \varphi_{ini}(\vec{\theta}_4) \text{ on } \Omega(\vec{\theta}_3), \\
\varphi &= g(\vec{\theta}_5) \text{ on } \partial\Omega(\vec{\theta}_3),
\end{aligned}
$$

Pomocí adjungovaných diferenciálních rovnic lze řešit úlohy tvaru

$$\min_{\vec{\theta}} L(\varphi(\vec{\theta})) \text{ pro } \vec{\theta} = (\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \vec{\theta}_4, \vec{\theta}_5).$$

Touto úlohou lze řešit:

- fitování modelů podle experimentálních dat
- řešení inverzní úlohy
- optimální řízení
- topologická optimalizace
- učení modelů - tzv. *physics from data*

Rychlé řešení pomocí GPU je téměř nezbytné.

Řešení inverzní rovnice vedení tepla

Rekonstrukce obrazových dat - Kateřina Škardová

- rovnice vedení tepla

$$
\begin{aligned}
\frac{\partial \varphi}{\partial t} - d\Delta\varphi &= 0 \text{ on } \Omega \times (0, T]\,, \\
\varphi\,|_{t=0} &= \varphi_{ini} \text{ on } \Omega, \\
\varphi &= g \text{ on } \partial\Omega,
\end{aligned}
$$

kde

$$
\Delta\varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2}.
$$

- ve zpracování obrazu se tato rovnice používá pro vyhlazování/rozmazávání obrazu

Řešení inverzní rovnice vedení tepla

Původní obrázek a dva rozmazané obrázky generované jako řešení rovnice vedení tepla. Difuzní koeficienty jsou $d_1 = 0.5$ a $d_2 = 0.75$.

Řešení inverzní rovnice vedení tepla

Rekonstrukci obrazu můžeme provést takto:

$$\min_{\theta} \int_{\Omega} \left(\varphi(x,y,T) - I(x,y)\right)^2 dx,$$

kde $I(x,y)$ je rozmazaný obraz a $\varphi(x,y,T)$ je řešení následující úlohy:

$$\begin{aligned} \frac{\partial \varphi}{\partial t} - d\Delta\varphi &= 0 \text{ on } \Omega \times (0,T], \\ \varphi\,|_{t=0} &= \varphi_{ini}(\vec{\theta}) \text{ on } \Omega, \\ \varphi &= g \text{ on } \partial\Omega. \end{aligned}$$

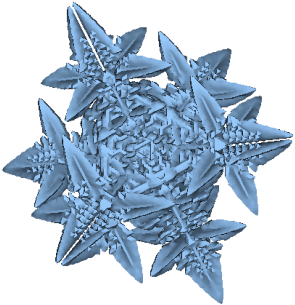
Řešení inverzní rovnice vedení tepla

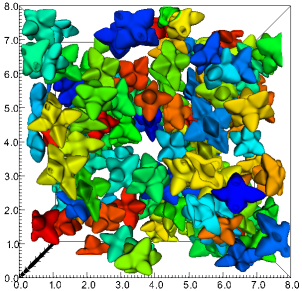
Rekonstrukce obrázku rozmazaného rovnicicí vedení tepla s difuzním koeficientem $d_1 = 0.5$.

Rekonstrukce obrázku rozmazaného rovnicicí vedení tepla s difuzním koeficientem $d_2 = 0.75$.

Optimální řízení pomocí okrajových podmínek

Řízení růstu krystalů - Pavel Strachota, Aleš Wodecki

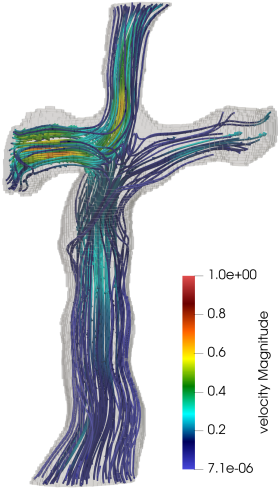




Optimální řízení pomocí okrajových podmínek

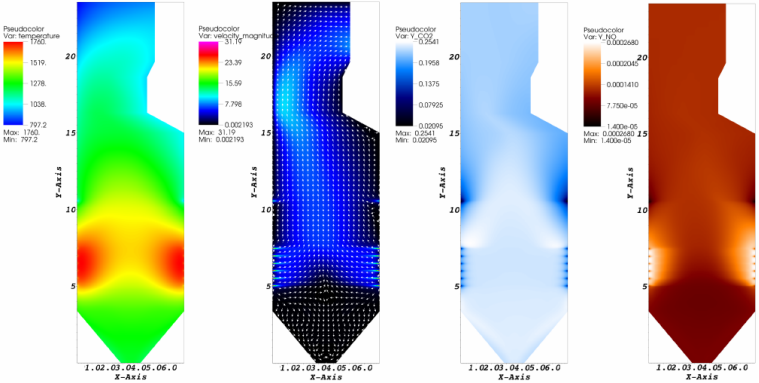
Fitování proudění v magnetické rezonanci - Radek Fučík, Pavel Eichler

- rychlostní pole z MR je velmi hrubé
- rekonstrukce pomocí LBM může být mnohem detailnější
- následně lze také optimalizovat chirurgicke zákroky



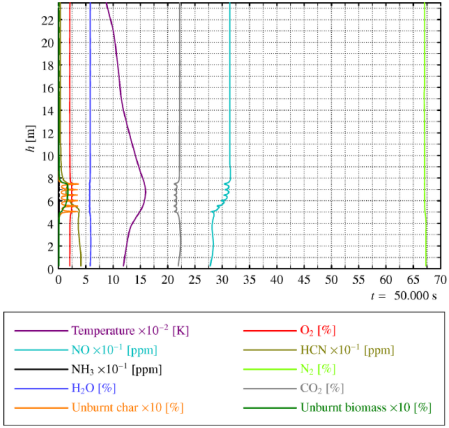
Optimální řízení pomocí okrajových podmínek

Spalování práškového uhlí - Pavel Strachota, Michal Beneš, Jakub Klinkovský, Jakub Solovský, Pavel Eichler



Optimální řízení pomocí okrajových podmínek

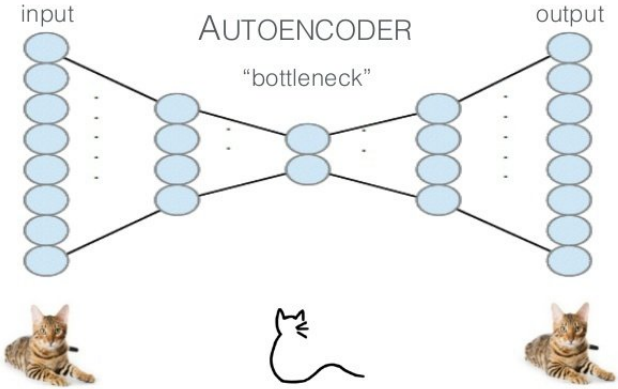
Spalování práškového uhlí:



Fitování/učení (zjenodušených) modelů

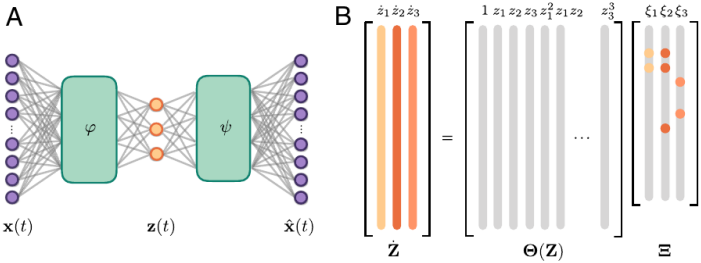
- modelování hoření v kotli je výpočetně extrémně náročné
- podle výsledků podrobného modelu se můžeme pokusit natrénovat jednodušší model, např. ODR, který bude popisovat pouze daný konkrétní kotel
- podobná úloha je např. srdeční perfuze
- nebo se můžeme pokusit učit model pouze podle experimentálních dat

Autoencodery a transformace



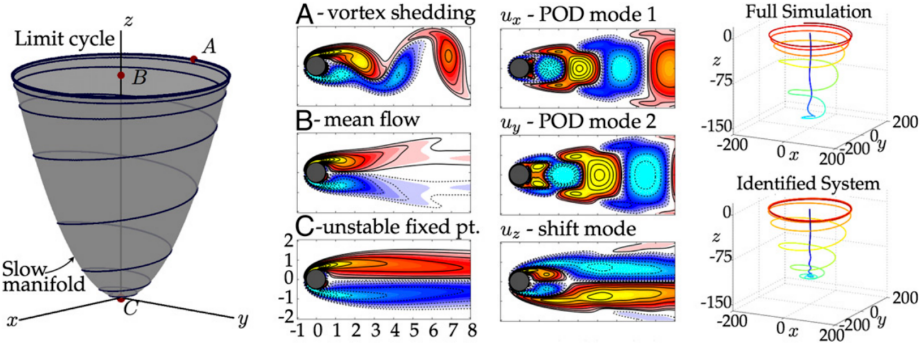
Autor: Sefik Ilkin Serengil, sefik.com

Autoencodery a transformace



K. Champion, B. Lusch, J. N. Kutz, S.L.Brunton, *Data-driven discovery of coordinates and governing equations*, PNAS, 2019.

Autoencodery a transformace



S.L.Brunton, J. L. Proctor, J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, PNAS, 2016.

Závěr

Další zdroje:

- GTC 2022 - Nvidia
  - Keynote with NVIDIA CEO Jensen Huang
- Steve Brunton - Youtube
- www.tnl-project.org

Děkuji za pozornost :-)