Solving the eikonal equation on GPU

Matouš Fencl

Supervisor: Ing. Tomáš Oberhuber, Ph.D.

CTU
Faculty of nuclear sciences and physical engineering

20. September 2020

Obsah

Eikonal equation

- Nonlinear partial differential equation
- assumption: continuous function $u_0(\boldsymbol{x})$

$$\begin{aligned} \Gamma &\equiv \{\boldsymbol{x} \in \mathbb{R}^n |\ u_0(\boldsymbol{x}) = 0\}, \\ ext(\Gamma) &\equiv \{\boldsymbol{x} \in \mathbb{R}^n |\ u_0(\boldsymbol{x}) > 0\}, \\ int(\Gamma) &\equiv \{\boldsymbol{x} \in \mathbb{R}^n |\ u_0(\boldsymbol{x}) < 0\}, \end{aligned}$$

- $\|\nabla u(\boldsymbol{x})\| = \frac{1}{v(\boldsymbol{x})}$, where $v(\boldsymbol{x}) > 0,\ \boldsymbol{x} \in \mathbb{R}^n$
- condition: $u(\boldsymbol{x}) = 0,\ \forall \boldsymbol{x} \in \Gamma$
- For $v(\boldsymbol{x}) = 1,\ \forall \boldsymbol{x} \in \mathbb{R}^n$ solution as signed distance function (SDF)

Solution

- signed distance function $d_\Gamma(\boldsymbol{x}) = \pm min\{\|\boldsymbol{x} - \boldsymbol{y}\|, \ \forall \boldsymbol{y} \in \Gamma\}$,
- where $+$ is chosen $\forall \boldsymbol{x} \in Ext(\Gamma)$, $- \ \forall \boldsymbol{x} \in Int(\Gamma)$
- solution can be calculated in every point from its neighbours
- solution is correct for two correct neighbours

Example of usage

- $\Gamma(t) = \{\boldsymbol{x} \in \mathbb{R}^n | u(\boldsymbol{x}) = t\}$ describes movement of borderline in time
- then $v(\boldsymbol{x})$ is velocity of spreading

Solution for cyrcle

- for Γ as cyrcle with radius of 1

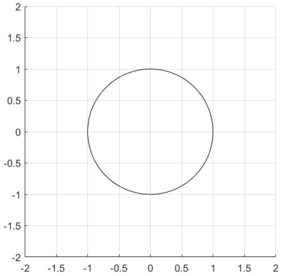


Figure: Initial condition cyrcle

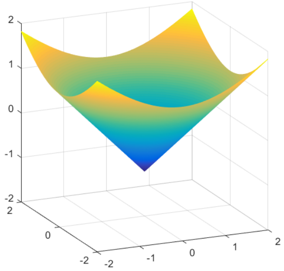


Figure: Solution SDF

Numerical description in 2D

- differentiations replaced with numerical differences

$$D^{-x}_{i,j}u = \frac{u_{i,j} - u_{i-1,j}}{h_x}, \qquad D^{+x}_{i,j}u = \frac{u_{i+1,j} - u_{i,j}}{h_x},$$
$$D^{-y}_{i,j}u = \frac{u_{i,j} - u_{i,j-1}}{h_y}, \qquad D^{+y}_{i,j}u = \frac{u_{i,j+1} - u_{i,j}}{h_y}$$

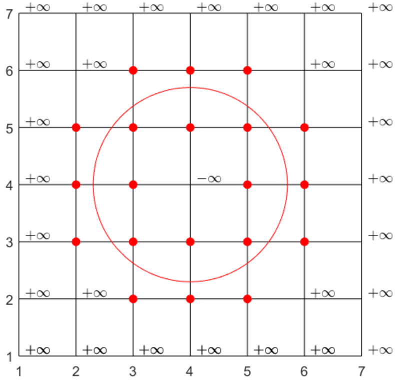
- Eikonal equation can be transfared into form

$$\sqrt{\max\{-D^{+x}_{i,j}u, D^{-x}_{i,j}u\}^2 + \max\{-D^{+y}_{i,j}u, D^{-y}_{i,j}u\}^2} = \frac{1}{v_{i,j}}, \tag{1}$$

$$(h_x^2 + h_y^2)u_{i,j}^2 + [-2(h_y^2 u^i_{min} + h_x^2 u^j_{min})]u_{i,j} + h_y^2(u^i_{min})^2 + h_x^2(u^j_{min})^2 - \frac{h_x^2 h_y^2}{v_{i,j}^2} = 0,$$

- where $u^i_{min} = min\{u_{i+1,j}, u_{i-1,j}\}$ and $u^j_{min} = min\{u_{i,j+1}, u_{i,j-1}\}$
- roots: $u_{i,j} = \frac{h_y^2 u^i_{min} + h_x^2 u^j_{min} \pm h_x h_y \sqrt{\frac{h_x^2 + h_y^2}{v_{i,j}^2} - (u^i_{min} - u^j_{min})^2}}{h_x^2 + h_y^2}$
- condition from discriminant: $|u^i_{min} - u^j_{min}| \leq \frac{\sqrt{h_x^2 + h_y^2}}{v_{i,j}}$

1. inicialization with $\pm\,\mathbf{inf}$
2. inicialization around Γ using linear approximation
3. recalculation of values in every point
   a. Fast Marching metoda $O(n \cdot ln(n))$
   b. Fast Sweeping metoda $O(n)$

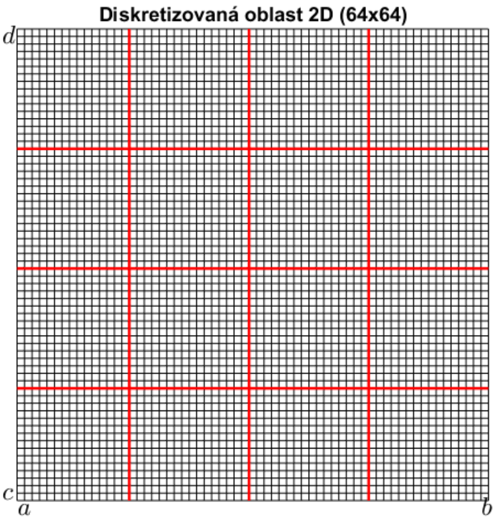


Implementace přepočítávání hodnot ve 2D

Algorithm 1 přepočítávání hodnot v bodech

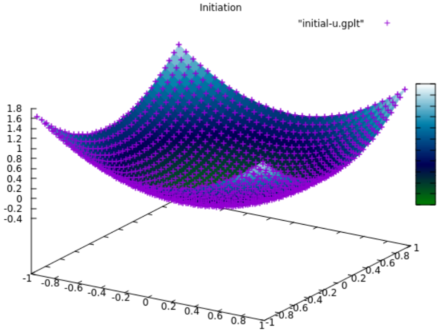
$u_{i,h_x}^{min} = min\{u_{i+1,j}, u_{i-1,j}\}$
$u_{j,h_y}^{min} = min\{u_{i,j+1}, u_{i,j-1}\}$
$[u_{1,h_1}, u_{2,h_2}] = argAbsMin\{u_{i,h_x}^{min},\ u_{j,h_y}^{min}\}$
if $|u_{1,h_1} + sgn(u_{0,i}) * h_1| < |u_{2,h_2}|$ **then**
  $u_{i,j} = absMin\{u_{i,j},\ u_{1,h_1} + sgn(u_{0,i}) * h_1\}$
else
  $$u_{i,j} = absMin\{u_{i,j}, \frac{h_y^2 u_{min}^i + h_x^2 u_{min}^j + sgn(u_{0,i,j}) h_x h_y \sqrt{\frac{h_x^2+h_y^2}{v_{i,j}^2} - (u_{min}^i - u_{min}^j)^2}}{h_x^2+h_y^2}\}$$
end if

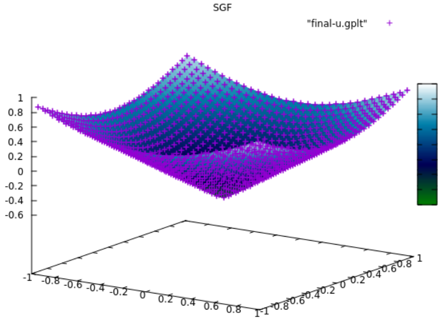
Ways to go through grid in paralel



Implementation

- Results for Γ as paraboloid:
- $u_0(x, y) = x^2 + y^2 - 0.75^2, \ x \in (-1, 1), \ y \in (-1, 1)$





Results in 2D

- paraboloid initial function: $u_0(x, y) = x^2 + y^2 - 0.75^2,\ x \in (-1, 1),\ y \in (-1, 1)$
- same space steps $h_x = h_y$
- $EOC = \dfrac{ln\left(\frac{||P_{h_1} y - u^{h_1}||}{||P_{h_2} y - u^{h_2}||}\right)}{ln\left(\frac{h_1}{h_2}\right)}$
- where $P_{h_1} y$ is projection y on numerical grid ω_{h_1}

Table: Table of EOC values for FSM with double precision of paraboloic initial function on CPU.

h	N×N	$\|\| . \|\|_{L_1}^h$ error	$\|\| . \|\|_{L_1}^h$ EOC	$\|\| . \|\|_{L_2}^h$ error	$\|\| . \|\|_{L_2}^h$ EOC
0.03125	64×64	0.0221759	**1.095**	0.0139008	**1.089**
0.015625	128×128	0.011458	**0.953**	0.0073657	**0.916**
0.0078125	256×256	0.00569208	**1.009**	0.00363781	**1.018**
0.00390625	512×512	0.00268996	**1.081**	0.00171859	**1.082**
0.001953125	1024×1024	0.00136002	**0.984**	0.000871464	**0.980**
0.0009765625	2048×2048	0.000685214	**0.989**	0.000438016	**0.992**
0.00048828125	4096×4096	0.000345106	**0.990**	0.000221592	**0.983**

Table: Comparition of executive times for 2D paraboloid function on GPU with double precision for methods: aktivních sousedů, šachovnice a 2 procesy aktivních sousedů s MPI, and its effectivness in comparision with aktivní sousedi method.

N^2	Aktivní sousedi	Šachovnice		Aktivní sousedi s MPI	
	čas	čas	urychlení	čas	urychlení
64^2	0.00159	0.00142	1.12x	0.00775	0.21x
128^2	0.00228	0.00176	1.30x	0.00957	0.24x
256^2	0.13202	0.03727	3.54x	0.01301	10.15x
512^2	0.01104	0.00725	1.52x	0.03836	0.29x
1024^2	0.03862	0.02370	1.63x	0.06273	0.62x
2048^2	0.19111	0.09762	1.96x	0.18762	1.02x
4096^2	1.30025	0.58810	2.21x	0.86949	1.50x

Wildfire

Figure: Mapa

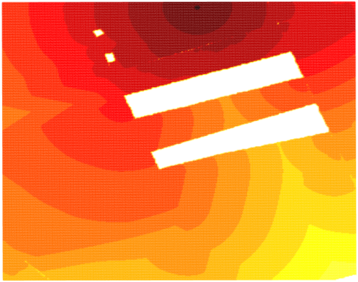


Figure: Šíření požáru v čase

Thank you for your attention!