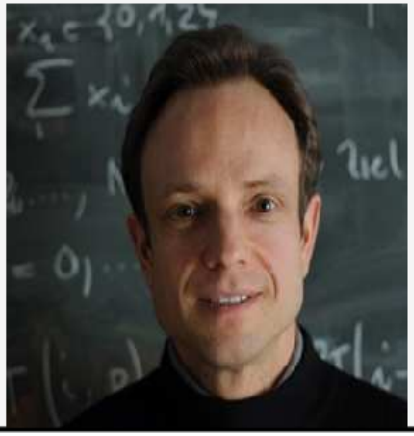
ETH Lecture 401-0674-00L Numerical Methods for Partial Differential Equations

Homework Problem Video Tutorial

Problem 8-2: Conservative finite-volume discretization based on Engquist-Osher numerical flux

Prof. R. Hiptmair, SAM, ETH Zurich

Date: May 20, 2019



Cauchy problem for 1D C.L.

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{on } \mathbb{R} \times]0,T[\,. \qquad u(x,0) = u_0(x) \quad \text{in } \mathbb{R}, \tag{8.2.1}$$

- Conservative FV in space
- Explicit Euler timestepping

2pt Engquist-Osher numerical flux

$$F_{EO}(v,w) := \tfrac{1}{2}(f(v) + f(w)) - \tfrac{1}{2}\int_v^w |f'(\xi)|\,\mathrm{d}\xi \,. \tag{8.2.2}$$

a) Consistency F_{EO}

Definition 8.3.3.5. Consistent numerical flux function

A numerical flux function $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}$ is consistent with the flux function $f : \mathbb{R} \mapsto \mathbb{R}$, if

$$F(u,\dots,u) = f(u) \quad \forall u \in \mathbb{R} \,.$$

$$F_{EO}(v,v) = f(v) \quad \checkmark$$

2

b) F_{EO} monotone

Definition 8.3.5.5. Monotone numerical flux function

A 2-point numerical flux function $F = F(v,w)$ is called monotone, if

F is an increasing function of its first argument v $(\forall w)$

and

F is a decreasing function of its second argument w $(\forall v)$.

Corollary 8.3.5.6. Simple criterion for monotone flux function

A continuously differentiable 2-point numerical flux function $F = F(v,w)$ is montone, if and only if

$$\frac{\partial F}{\partial v}(v,w) \geq 0 \quad \textit{and} \quad \frac{\partial F}{\partial w}(v,w) \leq 0 \quad \forall (v,w)\,. \tag{8.3.5.7}$$

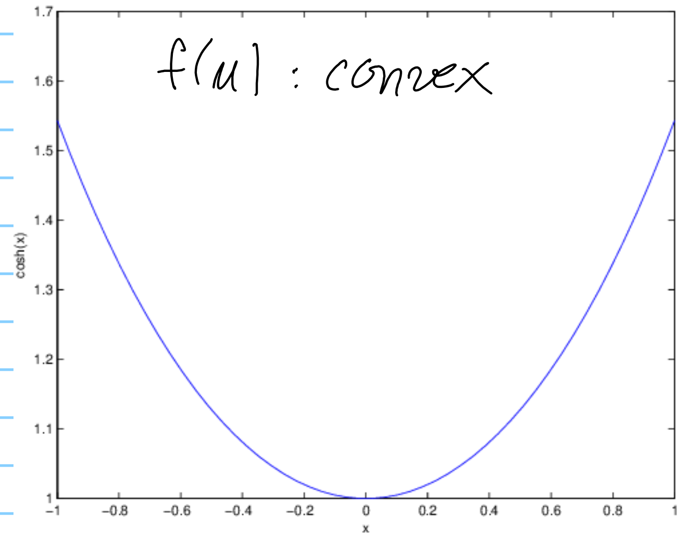
use fundamental thm. of calculus
& resolve $|\cdot|$

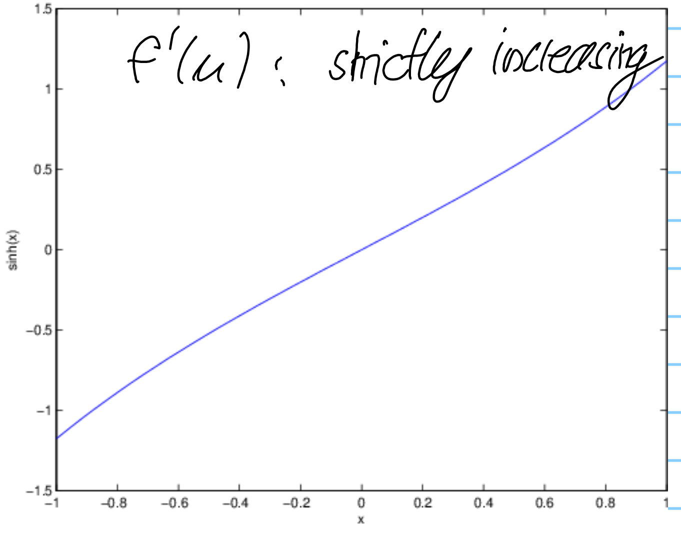
$$F_{EO}(v,w) := \tfrac{1}{2}(f(v)+f(w)) - \tfrac{1}{2}\int_v^w |f'(\xi)|\,d\xi\,. \tag{8.2.2}$$

$$\frac{\partial F_{EO}}{\partial v}(v,w) = \tfrac{1}{2}f'(v) + \tfrac{1}{2}|f'(v)| = \begin{cases} f'(v) & \text{, if } f'(v) \geq 0\,, \\ 0 & \text{, if } f'(v) < 0 \end{cases} \geq 0\,,$$

$$\frac{\partial F_{EO}}{\partial w}(v,w) = \tfrac{1}{2}f'(w) - \tfrac{1}{2}|f'(w)| = \begin{cases} 0 & \text{, if } f'(w) \geq 0\,, \\ f'(w) & \text{, if } f'(w) < 0 \end{cases} \leq 0\,.$$

Now: $f(u) = \cosh(u) \Rightarrow f'(u) = \sinh(u)$





d) If u_0 is supported in $[0,1]$ and $-1 \leq u_0(x) \leq 1$ for all $x \in \mathbb{R}$, find the maximal possible support of $u(\cdot,t)$ for time $t > 0$, where $u(x,t)$ solves (8.2.1).

Maximal slopes of characteristics [f' increasing]

$S_{max} = f'(u_{max}) = \sinh(1)$

$S_{min} = f'(u_{min}) = -\sinh(1)$

$\in \{1,-1\}$ by comparison theorem

$\triangleright$ $\quad \mathrm{supp}(u(\cdot,t)) \subset [-\sinh(1)t,\ 1+\sinh(1)t]$

③ e) Implementation of F_{EO}

Use: $f'(u) \geq 0 \iff u \geq 0$

$$F_{EO}(v,w) := \tfrac{1}{2}(f(v)+f(w)) - \tfrac{1}{2}\int_v^w |f'(\xi)|\,\mathrm{d}\xi \,. \tag{8.2.2}$$

Idea: Split domain of integration into negative/positive parts & resolve modulus

$$F_{EO}(v,w) = \tfrac{1}{2}(f(v)+f(w)) - \frac{1}{2}\cdot\begin{cases} f(w)-f(v) & \text{, if } v,w \geq 0\,, \\ f(v)+f(w)-2f(0) & \text{, if } v<0,\ w\geq 0\,, \\ 2f(0)-f(v)-f(w) & \text{, if } w<0,\ v\geq 0\,, \\ f(v)-f(w) & \text{, if } v,w<0\,. \end{cases}$$

$$= \begin{cases} f(v) & \text{, if } v,w \geq 0\,, \\ f(0) & \text{, if } v<0,\ w\geq 0\,, \\ f(v)+f(w)-f(0) & \text{, if } w<0,\ v\geq 0\,, \\ f(w) & \text{, if } v,w<0\,. \end{cases}$$

g)

Determine the CFL-condition introduced in [Lecture → Def. 8.4.2.10] for the conservative finite volume method for (8.2.1) introduced above, if it is known that $A \leq u_0(x) \leq B$ for all $x \in \mathbb{R}$, with $A, B \in \mathbb{R}$, $A < B$.

↳ $u(x,t) \in [A,B]$

Maximal speed of propagation

f' monotone: $\dot{s} = \max\{|f'(A)|, |f'(B)|\}$

$\boxed{\text{2-point flux, explicit Euler}}$ $\Big\} \Rightarrow$ Stencil width $m=1$ for $\mathcal{H}_h$

▷ $\tau \leq h/\dot{s}$

h) Fully discrete evolution

$$\mu_j^{(k+1)} = \mu_j^{(k)} - \tfrac{\tau}{h}\left(F_{EO}(\mu_j^{(k)},\mu_{j+1}^{(k)}) - F(\mu_{j-1}^{(k)},\mu_j^{(k)})\right)$$

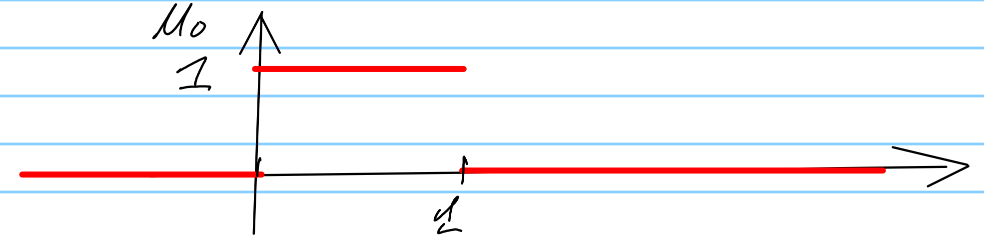
+ truncation to finitely many j-s.

i)

Use the function `solveCP()` to solve (8.2.1) for initial data

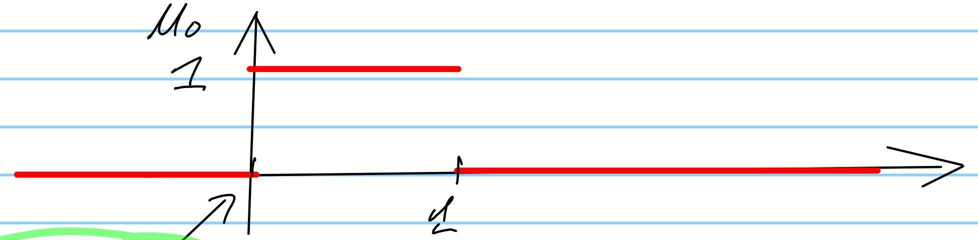
$$u_0(x) = \begin{cases} 1 & \text{for } 0 < x \leq 1, \\ -1 & \text{elsewhere,} \end{cases}$$ Box function

and final time $T = 1$. Use $[-1.2, 2.2]$ as computational interval and $N = 100$. The cell avarages of u_0 can be approximated by its values at the cell centers. To that end extend the `main()` function so that it outputs the approximate dual cell averages at final time to a file `ufinal.csv`. This should be read by a suitable MATLAB or PYTHON script plotting the numerical approximation of $u(x,T)$ on $[-1.2, 2.2]$. Alternatively you may use MATHGL to create a plot in `ufinal.eps` directly from the C++ code.



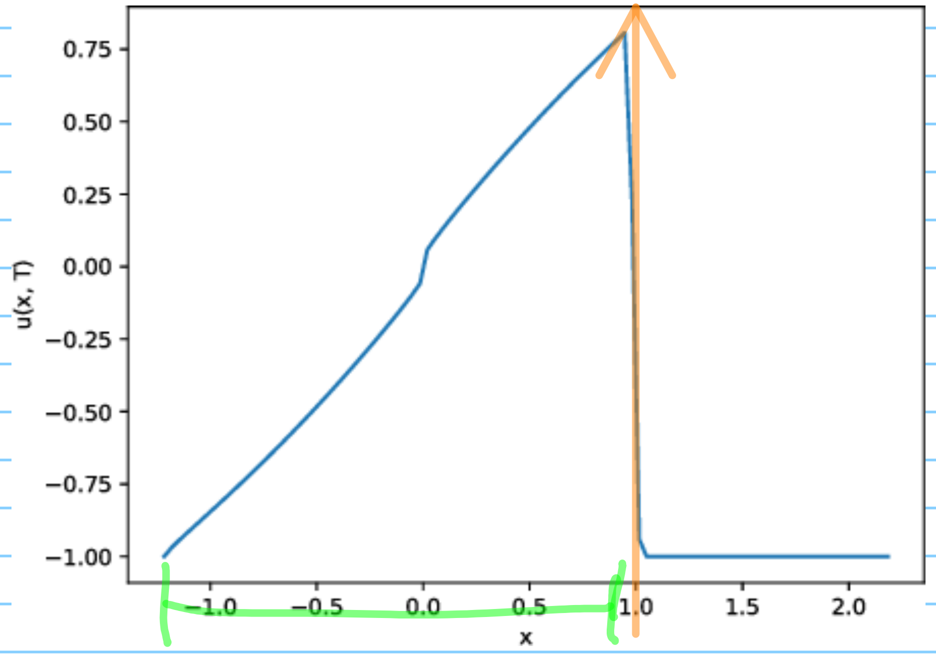
(4)

f convex !



gives rise to a rarefaction fan!

triggers shock



5