Exercise set 8

Numerical Methods for Hyperbolic Partial Differential Equations
IMATH, FS-2020
Lecturer: Dr. Philipp Öffner
Teaching Assistant: Davide Torlo

Problem 8.1 Lax Wendroff scheme (3pts)

Consider the Lax Wendroff scheme

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}\left(f(u_{j+1}^n) - f(u_{j-1}^n)\right) + \frac{\lambda^2}{2}\left(a_{j+1/2}\Delta f_{j+1/2} - a_{j-1/2}\Delta f_{j-1/2}\right), \tag{1}$$

with

$$a_{j+1/2} = \begin{cases} \frac{f(u_j)-f(u_{j+1})}{u_j-u_{j+1}} & \text{if } u_j \neq u_{j+1},\\ f'(u_j) & \text{if } u_j = u_{j+1}, \end{cases} \tag{2}$$

with the usual notation $\Delta p_{j+1/2} = p_{j+1} - pj$ and $\lambda = \Delta t/\Delta x$. We have shown in previous exercises that it is a second order scheme which is not TVD, hence, it is not monotone and, for linear equations, it is not monotonicity preserving.

1. Prove that the scheme is von Neumann stable under suitable CFL conditions for a linear advection problem.

Solution

$U_t + cU_x = 0 \qquad U_k = e^{\alpha t} e^{ixk} \qquad a_{j+1/2} = \frac{cU_j - cU_{j+1}}{U_j - U_{j+1}} = c$

$e^{\alpha\Delta t} U_j^n = U_j^n - \frac{\lambda c}{2}\left(e^{i\Delta x k} - e^{-i\Delta x k}\right) U_j^n + \frac{\lambda^2 c}{2}\left(c(U_{j+1}^n - U_j^n) - c(U_j^n - U_{j-1}^n)\right)$

$e^{\alpha\Delta t} = 1 - \beta i \sin(\theta) + \frac{\beta^2}{2}\left(e^{i\Delta x k} - 2 + e^{-i\Delta x k}\right)$

$\beta = \lambda c \qquad \theta = k\Delta x$

$e^{\alpha\Delta t} = 1 - \beta i \sin(\theta) + \beta^2(\cos(\theta) - 1)$

$(1-\cos)(1+\cos)$

$|e^{\alpha\Delta t}|^2 = (1 - \beta^2(1-\cos(\theta)))^2 + \beta^2 \sin^2(\theta)$

$= 1 - 2\beta^2(1-\cos(\theta)) + \beta^4(1-\cos(\theta))^2 + \beta^2(1-\cos(\theta)^2)$

$|e^{\alpha\Delta t}|^2 \leq 1$?

$\iff$

$\beta^2(1-\cos(\theta))\left[-2 + \beta^2(1-\cos\theta) + 1 + \cos(\theta)\right] \leq 0$?

$\beta^2(1-\cos(\theta))\left[\beta^2(1-\cos(\theta)) - (1-\cos(\theta))\right] \leq 0$

$\underbrace{\beta^2}_{\geq 0}\underbrace{(1-\cos(\theta))^2}_{\geq 0}\left[\beta^2 - 1\right] \leq 0$? $\iff \beta^2 - 1 \leq 0 \quad \boxed{\beta^2 \leq 1}$

VON NEUMANN STABLE $\Leftarrow |c\lambda| \leq 1$ CFL $\boxtimes$

2. Write the numerical flux in the conservative form for (1).

Solution

$f_{LW}^{num}(U_j, U_{j+1}) = \frac{f(U_j) + f(U_{j+1})}{2} - \frac{\lambda}{2}\left(a_{j+1/2}\Delta f_{j+1/2}\right)$

$U_j^{n+1} = U_j^n - \lambda\left(f_{LW}^{num}(U_j, U_{j+1}) - f_{LW}^{num}(U_{j-1}, U_j)\right)$. $\boxtimes$

3. Code its numerical flux in the code of the Exercise Set 6 (you can use the one provided as solution).

Solution

```
function [fNum] = numericalFlux(scheme,f,u,v,extra)
% encode in this function different numerical fluxes
switch scheme
    case "Lax Friedrichs"
        % extra should be lambda=dt/dx
        lam=extra{1};
        fNum=(f(u)+f(v))/2-(v-u)/lam/2;
    case "Lax Wendroff"
        % extra should be lambda=dt/dx
        lam=extra{1};
        df=extra{2};
        idxn= u==v;
        idx=logical(1-idxn);
        a=zeros(size(u));
        a(idx)=(f(u(idx))-f(v(idx)))./(u(idx)-v(idx));
        fNum=(f(u)+f(v))/2-lam/2*a.*(f(v)-f(u));
    case "Rusanov"
        % extra should be f'
        df=extra{1};
        fNum=(f(u)+f(v))/2-max(abs(df(u)),abs(df(v))).*(v-u)/2;
    case "Godunov"
        % extra should be omega the unique local minimum of f
        omega=extra{1}*ones(size(u));
        fNum=max(f(max(u,omega)),f(min(v,omega)));
    case "Roe"
        % extra should be empty
        idxs=u==v;
        idx=logical(1-idxs);
        fNum=f(u);
        A=(f(u(idx))-f(v(idx)))./(u(idx)-v(idx));
        fNum(idx)=f(u(idx)).*(A>=0)+f(v(idx)).*(A<0);
    case "EO" %Engquist--Osher
        % extra should be omega the unique local minimum of f
        omega=extra{1}*ones(size(u));
        fNum=f(max(u,omega))+ f(min(v,omega));
end

end
```

Listing 1: NumericalFlux.m

$\frac{\Delta t}{\Delta x} = \lambda \qquad df$

$a_{j+1/2} = \frac{f(U_j) - f(U_{j+1})}{U_j - U_{j+1}}$ WHEN $U_j \neq U_{j+1}$

Problem 8.2 Entropy (7pts)

Consider the Burgers' equation

$$\partial_t u + \partial_x f(u) = 0, \quad f(u) = \frac{u^2}{2}, \quad x \in [-2,2],\, t \in [0,1], \tag{3}$$

with periodic boundary conditions.

1. Write Kruzkov's entropy U for $\ell = 0$ for Burgers' equation and its entropy flux q.

Solution

$U(u) = |u - \ell| = |u| \qquad \boxed{U' f' = q'}$

$q(u) = \text{sign}(u-\ell)\cdot(f(u) - f(\ell)) = \text{sign}(u)\cdot f(u) = \begin{cases} f(u) & u > 0 \\ -f(u) & u < 0 \end{cases}$

$\text{sign}(u)\cdot f'(u) = f'(u)\cdot\text{sign}(u)$ a.e.

BURGERS'

$U(u) = |u| \qquad q(u) = \text{sign}(u)\frac{u^2}{2} = \frac{u|u|}{2}$

$\text{sign}(u)\cdot u = |u|$

$|u| = |u|$

2. Consider then the entropy

$$U(u) = \frac{u^2}{2}, \tag{4}$$

compute the entropy flux in this case.

Solution

$U' f' = u \cdot u = q' \qquad q(u) = \int_{U_0}^{u} u^2\, du = \frac{u^3}{3} + \cancel{C} = \frac{U^3}{3}$

BREAK → 11.10

3. Consider the Lax–Friedrichs scheme, defined by the numerical flux

$$f^{num}(u,v) = \frac{f(u)+f(v)}{2} - \frac{1}{2\lambda}(v-u). \tag{5}$$

Using Crandall–Majda's lemma, write the related entropy numerical flux for Kurzkov's entropy for a general conservation law with flux f.

Solution

$U(U_j^{n+1}) \leq U(U_j^n) - \lambda\left(q^{num}(U_j^n, U_{j+1}^n) - q^{num}(U_{j-1}^n, U_j^n)\right)$

$q^{num}(a,b) = f^{num}(a\vee\ell, b\vee\ell) - f^{num}(a\wedge\ell, b\wedge\ell) =$

$= \frac{f(a\vee\ell) + f(b\vee\ell)}{2} - \frac{b\vee\ell - a\vee\ell}{2\lambda} - \frac{f(a\wedge\ell) + f(b\wedge\ell)}{2} + \frac{b\wedge\ell - a\wedge\ell}{2\lambda} = (*)$

NOTE:

1) $f(a\vee\ell) - f(a\wedge\ell) = \text{sign}(a-\ell)\cdot(f(a) - f(\ell)) = q(a)$

2) $a\vee\ell - a\wedge\ell = |a-\ell| = U(a)$

$(*) = \frac{q(a) + q(b)}{2} - \frac{U(b) - U(a)}{2\lambda}$

4. Write the numerical entropy inequality for Lax–Friedrichs in terms of the entropy U and of the entropy flux q.

Solution

$U(U_j^{n+1}) \leq U(U_j^n) - \lambda\left(\frac{q(U_{j+1}^n) - q(U_{j-1}^n)}{2} - \frac{U(U_{j+1}^n) - 2U(U_j^n) + U(U_{j-1}^n)}{2\lambda}\right)$

$U(U_j^{n+1}) \leq \frac{U(U_{j+1}^n) + U(U_{j-1}^n)}{2} - \frac{\lambda}{2}\left(q(U_{j+1}^n) - q(U_{j-1}^n)\right)$ $\boxtimes$

$\boxed{5}$

$\sum_j U(U_j^{n+1}) \leq \sum_j U(U_j^n) - \lambda \underbrace{\sum_j \left(q^{num}(U_j^n, U_{j+1}^n) - q^{num}(U_{j-1}^n, U_j^n)\right)}_{=0}$

$\underbrace{\sum_j U(U_j^{n+1}) \leq \sum_j U(U_j^n)}$

WHAT WE CHECK IN THE CODE

5. Check numerically that Kruzkov's entropy and (4) are diminishing for Lax–Friedrichs and that are not always diminishing for Lax–Wendroff (2 entropies, 2 fluxes, 2 initial conditions). Use as test examples for Burgers' equation on $[-2,2]$ with periodic BC and final time 1 the following IC

$$u_0(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{else;} \end{cases} \tag{6}$$

and

$$u_0(x) = 0.2\cos(\pi x). \tag{7}$$

Submit the code and an adequate number of plots (or a script that automatically generates all the plots). **Hint**: the entropy does not diminish locally, but globally... Find a good way of measuring the loss of total entropy.

Solution

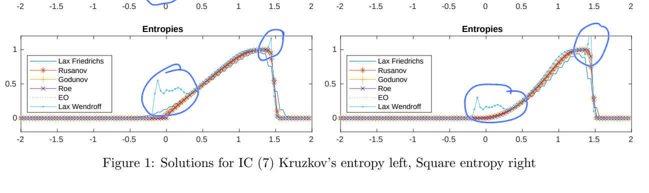

Figure 1: Solutions for IC (7) Kruzkov's entropy left, Square entropy right

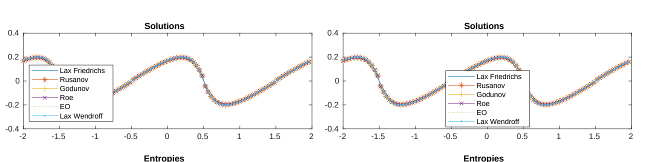

Figure 2: Solutions for IC (6) Kruzkov's entropy left, Square entropy right

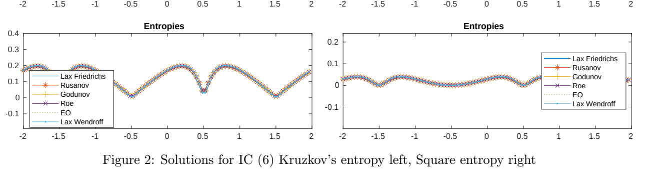

Figure 3: Entropy decay for IC (7) Kruzkov's entropy left, Square entropy right

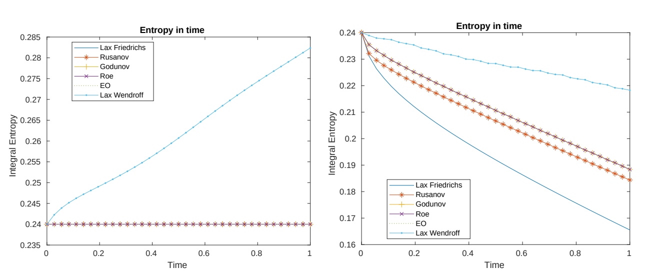

Figure 4: Entropy decay for IC (6) Kruzkov's entropy left, Square entropy right

```
function [u,x,t,ent] = runScheme(model, solver)
N=solver.Nx;
x=linspace(model.a,model.b,solver.Nx+1);
dx=x(2)-x(1);
maxdfu=max(abs(model.df(model.u0(x))));
dt=dx*solver.CFL/maxdfu;
Nt=ceil(model.T/dt);
dt=model.T/Nt;
t=linspace(0,model.T,Nt+1);
%u(j,n+1)= H(u(j-1,n),u(j,n),u(j+1,n))
if model.BC=="periodic"
    j=1:N;
    jR=[2:N,1];
    jL=[N, 1:N-1];
    u=zeros(Nt+1,N);
    ent=zeros(Nt+1,N);
    x=x(j);
elseif model.BC=="dirichlet"
    j=[2:N];
    jL=[1:N-1];
    jR=[3:N+1];
    x=x(1:N+1);
    u=zeros(Nt+1,N+1);
    ent=zeros(Nt+1,N+1);
    u(:,1)=model.u0(x(1));
    u(:,N+1)=model.u0(x(N+1));
end

u(1,:)=model.u0(x);
ent(1,:)=model.entropy(u(1,:));
extra=defineExtraScheme(solver.scheme,dt/dx,model.df);
for kt=2:Nt+1
    u(kt,j)=u(kt-1,j) - dt/dx*(...
        numericalFlux(solver.scheme,model.f,u(kt-1,j),u(kt-1,jR),extra)...
        -numericalFlux(solver.scheme,model.f,u(kt-1,jL),u(kt-1,j),extra));
    ent(kt,:)=model.entropy(u(kt,:));
end

end


function extra=defineExtraScheme(scheme,lam,df)
switch scheme
    case "Lax Friedrichs"
        extra={lam};
    case "Lax Wendroff"
        extra={lam, df};
    case "Rusanov"
        extra={df};
    case {"Godunov","EO"}
        extra={0}; %only Burgers
    case "Roe"
        extra={};
end

end
```

Listing 2: runScheme.m

$\Delta t \leq \frac{\Delta x}{\max|f'(U)|}$

a b

1 2 N N+1

u 0

DIRI

u

```
model.f=@(u) u.^2/2;
model.df=@(u) u;%ones(size(u));
model.T=1;
model.a=-2;
model.b=2;
uL=0;%1;%-1
uR=1;%0;%1
model.u0=@(x) (x<1).*(x>0);%0.2* cos(pi*x);% uL*(x<0)+uR*(x>=0); %cos(pi*x);%cos(pi*x);
model.BC="periodic";%"dirichlet";%"periodic";

model.entropy = @(u)  abs(u);%u.^2;%abs(u); %u.^2/2; %@(u)
model.eFlux = @(u) sign(u).*f(u); % @(u) u.^3/3;

%% run lxf scheme
solver.scheme="Lax Wendroff";%"Lax Wendroff";%"Rusanov";%"Godunov";%"Roe";%Rusanov";%"EO";%"
    Lax Friedrichs"

solver.Nx = 100;
solver.CFL = 0.7;

[u,x,t,ent]= runScheme(model, solver );

%% plot evolution
plot_evolution=0;

dx = x(2)-x(1);
if(plot_evolution)
fig=figure(1);
    for n=[1:ceil(numel(t)/20):numel(t),numel(t)]
        %
        figure(1)
        subplot(211)
        plot(x,u(n,:))
        legend('numerical','Location','SE')
        xlabel x
        ylabel u
        title(sprintf('%s, CFL=%g, t=%g',solver.scheme,solver.CFL,t(n)))
        ylim([min(u,[],'all'),max(u,[],'all')])

        subplot(212)
        plot(x,ent(n,:))
        legend('numerical','Location','SE')
        xlabel x
        ylabel u
        title(sprintf('Entropy %s, CFL=%g, t=%g',solver.scheme,solver.CFL,t(n)))
        ylim([min(ent,[],'all')-0.1,max(ent,[],'all')+0.1])



        %
        drawnow
        pause(.02)
    end
end

%%
schemes=["Lax Friedrichs";"Rusanov";"Godunov";"Roe";"EO";"Lax Wendroff"];
styles=["-","*-","+-","x-",":",".-"]

clear u x t ent

for k=1:length(schemes)
    solver.scheme=schemes(k);
    [u{k},x{k},t{k},ent{k}]= runScheme(model, solver);
end

figure()
for k=1:length(schemes)
    scheme=schemes(k);
    subplot(211)
    plot(x{k},u{k}(end,:),styles(k),'DisplayName',scheme)
    hold on
    subplot(212)
    plot(x{k},ent{k}(end,:),styles(k),'DisplayName',scheme)
    hold on
end

subplot(211)
legend('Location','best')
title('Solutions')
ylim([min(u{1},[],'all')-0.2, max(u{1},[],'all')+0.2])

subplot(212)
legend('Location','best')
title('Entropies')
ylim([min(ent{1},[],'all')-0.2, max(ent{1},[],'all')+0.2])


figure()
for k=1:length(schemes)
    scheme=schemes(k);
    plot(t{k},sum(ent{k},2)/solver.Nx,styles(k),'DisplayName',scheme)
    hold on
end
title('Entropy in time')
xlabel('Time')
ylabel('Integral Entropy')
legend('Location','best')
```

$\approx \sum_j U(U_j^n) \quad \forall n$

Listing 3: testEntropy.m

Organiser: Davide Torlo, Office: home (davide.torlo@math.uzh.ch)

Published: Apr 23, 2020

Due date: Apr 30, 2020, h10.00 (use the upload tool of my.math.uzh.ch, see wiki.math.uzh.ch/public/student_upload_homework or if you have troubles send me an email).