Solution set 3:

Conservative methods

Exercise 3.1 **(a)** See figures generated by the code attached at the end of the solution manual. **(b)** The numerical solution converges to the function $u(x,t) = u(x,0)$. This shock is not a weak solution of the conservation law. To see this take any $0 \leq T_1 < T_2$. Then

$$\int_{-1}^{1} u(x,T_2)\,dx - \int_{-1}^{1} u(x,T_1)\,dx = 0 \tag{1}$$

while

$$\int_{T_1}^{T_2} f(u(1,t))\,dt - \int_{T_1}^{T_2} f(u(-1,t))\,dt = \int_{T_1}^{T_2} f(0)\,dt - \int_{T_1}^{T_2} f(1)\,dt = \frac{T_2 - T_1}{2} > 0 \tag{2}$$

(c) See figures generated by the code attached at the end of the solution manual. **(d)** The entropy solution is given by

$$u(x,t) = \begin{cases} 1.2 & x < 0.8t \\ 0.4 & 0.8t < x \end{cases} \tag{3}$$

Notice that in this example, the numerical solution simulates a shock that travels in the correct direction, but at a speed different than the actual shock speed. The important thing to note here is that in the two examples presented in the exercise, the upwind inspired method, which can be proven to work well for smooth solutions, returned non-solutions! Thus, numerical methods that are both consistent and stable on problems with smooth solutions may actually converge to non-solutions when applied to nonlinear problems with discontinuities in the solution. Trouble! This is the motivation for using conservative methods, when applying a conservative method we are guaranteed that the method converge to a weak solution to the problem.

Exercise 3.2 **(a)** See figures generated by the code attached at the end of the solution manual. **(b)** Yes. **(c)** Yes. **(d)** A scheme is conservative if it can be written as

$$U_j^{n+1} = U_j^n - \frac{k}{h}\left[F\left(U_j^n, U_{j+1}^n\right) - F\left(U_{j-1}^n, U_j^n\right)\right] \tag{4}$$

where F is a smooth function that satisfies $F(u,u) = f(u)$. The Lax-Friedrichs scheme is normally written as

$$U_j^{n+1} = \frac{1}{2}\left(U_{j+1}^n + U_{j-1}^n\right) - \frac{k}{2h}\left(f\left(U_{j+1}^n\right) - f\left(U_{j-1}^n\right)\right) \tag{5}$$

We can rewrite is as

$$\begin{aligned} U_j^{n+1} - U_j^n =& \frac{1}{2}\left[\left(U_{j+1}^n - U_j^n - \frac{k}{h} f\left(U_{j-1}^n\right)\right) - \left(U_j^n - U_{j-1}^n - \frac{k}{h} f\left(U_{j-1}^n\right)\right)\right] \\ =& \frac{k}{h}\left[\left(\frac{h}{2k}\left(U_{j+1}^n - U_j^n\right) - \frac{1}{2} f\left(U_{j+1}^n\right)\right) - \left(\frac{h}{2k}\left(U_j^n - U_{j-1}^n\right) - \frac{1}{2} f\left(U_{j-1}^n\right)\right)\right] \end{aligned}$$

It might be tempting to choose $F(u,v) = \frac{h(v-u)}{2k} - \frac{f(v)}{2}$, however this numerical flux would not satisfy consistency $F(u,u) = f(u)$. Instead, conservation is achieved by taking

$$F(u,v) = \frac{h}{2k}(v-u) - \frac{1}{2}\left(f(v) + f(u)\right) \tag{6}$$

(f) No, we are not in general guaranteed that this will be the case.

Exercise 3.3 **(a)** If it can be written in the form Eq. 4. **(b)** We do not have to worry about the scheme returning non-solutions, weak-solutions can be guaranteed. **(c)** No, not in general.

```
% solution03.m clear all,clc
% This script was written for EPFL MATH459, Numerical Methods for Conservation Laws and tested
% with Octave 3.6.4.
% The inviscid burgers equation is being solved using two different sets of initial condtions
% using an upwind inspired scheme and the generalized lax-friedrichs scheme. The first method
% is non-conservative, the second is conservative.

% Discretization
h = 0.01;
k = 0.5*h;
X = -0.5:h:1;
nX = numel(X);
T = 0:k:0.5;
nT = numel(T);

% Solve with initial data Eq. 3
u = zeros(nX,1);
u(find(X<0),1) = 1;
U1 = u;U2 = u;
f1 = figure(1)
for i = 1:nT
    % Calculate next timestep using the upwind inspired numerical scheme
    U1(2:end) = U1(2:end) - (k/h) * U1(2:end) .* ( U1(2:end) - U1(1:end-1) );
    % Calculate next timestep using the generalization of the Lax-Friedrichs method
    U2(2:end) = 0.5 * ( U2(1:end-1) + [U2(3:end);0] )- (0.25*k/h) * ...
                            ( [U2(3:end);0].^2 - U2(1:end-1).^2 );
    % The true solution, shock speed calculated using Huginiot jump condition
    u(find(X<0.5*T(i)),1) = 1;
    % Visualize the numerical approximation and the correct solution
    plot(X,U1(:),'-b',X,U2(:),'-r',X,u(:),'-k');
    ylim([-0.25 1.75]);xlim([-0.5 1]);
    set(gca,'XTick',-0.5:0.5:1);set(gca,'YTick',0:0.5:1.5)
    grid on;
    title('Exercise 3.1');
    legend('Upwind inspired','Generalized Lax-Friedrichs','True solution')
    drawnow
end

% Solve with initial data Eq. 4
u = zeros(nX,1);
u(find(X<0),1) = 1.2;
u(find(X>=0),1) = 0.4;
U1 = u; U2 = u;
f2 = figure(2)
for i = 1:nT
    % Calculate next timestep using the upwind inspired numerical scheme
    U1(2:end) = U1(2:end) - (k/h) * U1(2:end) .* ( U1(2:end) - U1(1:end-1) );
    % Calculate next timestep using the generalization of the Lax-Friedrichs method
    U2(2:end) = 0.5 * ( U2(1:end-1) + [U2(3:end);0.4] )- (0.25*k/h) * ...
                          ( [U2(3:end);0.4].^2 - U2(1:end-1).^2 );
    % The true solution, shock speed calculated using Huginiot jump condition
    u(find(X<(4/5)*T(i)),1) = 1.2;
    u(find(X>=(4/5)*T(i)),1) = 0.4;
    % Visualize the numerical approximation and the correct solution
    plot(X,U1(:),'-b',X,U2(:),'-r',X,u(:),'-k');
    ylim([-0.25 1.75]);xlim([-0.5 1]);
    set(gca,'XTick',-0.5:0.5:1);
    set(gca,'YTick',0:0.5:1.5);
    grid on;title('Exercise 3.2');
    legend('Upwind inspired','Generalized Lax-Friedrichs','True solution')
    drawnow
end
```