Num. #4: 1D hyperbolic PDEs system - Correction

The programs are written with the MATLAB software.
For the exercise, the following functions are needed

- **Rusanov method** :

```
%% Rusanov method
%% f defines the system of conservation laws and
%% specrad is the spectral radius of the jacobian
%% d is the dimension of the system
function[vfinal]=Rusanov(T,L,dx,vinit,f,specrad,d)
  %% Initial datum - We calculate on N-1 points
  v=vinit(1:end-1,:);
  %%Rusanov method
 %% for i=1:Nt
 Tsimul=0;
 while  Tsimul<T
      %% Neumann boundary conditions
      %% Computation of vectors up(i)=u_{i+1}, um(i)=u_(i-1)
      if d==3
          vp=[v(2:end,:);v(end,1),-v(end,2),v(end,3)];
          vm=[v(1,1),-v(1,2),v(1,3);v(1:end-1,:)];
      elseif d==2
          vp=[v(2:end,:);v(end,1),-v(end,2)];
          vm=[v(1,1),-v(1,2);v(1:end-1,:)];
      end
      %% computation of the velocities
       a=max(specrad(v),specrad(vp));
      am=max(specrad(vm),specrad(v));
      %% Time step
      dt=0.95*dx/max(specrad(v));
      Tsimul=Tsimul+dt;
       %% computation of flux
      Fp=(f(v)+f(vp))/2-diag(a)*(vp-v)/2;
      Fm=(f(vm)+f(v))/2-diag(am)*(v-vm)/2;
      v=v-dt/dx*(Fp-Fm);
end
   if d==3
     vfinal=[v;v(end,1), -v(end,2),v(end,3)];
   elseif d==2
     vfinal=[v;v(end,1), -v(end,2)];
```

```
end
```

Exercise

1. Compute the eigenvalues of the jacobian of systems (2) and (4).
2. Implement the Rusanov (or Local Lax-Friedrichs) scheme.
3. Test it for the shallow-water system (2) with initial data (3a) and (3b) on the domain $[0,5]$ with $\Delta x = 0.01$. The time step will be adapted at every step according to the stability condition. We will use Neumann boundary conditions.

```
clear;
clf;
g=9.81;
%% Space discretization
L=5;
dx=0.01;
space=(0:dx:L)';
%% Time discretization
T=1;
%% Initial datum 1 for height
%hinit=exp(-(space-2).^2/0.1);
%% Initial datum 2 for height
space1=space(space<2.5);
space2=space(space>=2.5);
hinit=[2*ones(size(space1));ones(size(space2))];
%% Initial datum for the velocity
uinit=zeros(size(space));
%% Function and spectral radius of jacobian for the Shallow-Water system
f=inline('[u(:,2),(u(:,2).^2)./u(:,1)+9.81*u(:,1).^2/2]','u');
specrad=inline('max(abs(u(:,2)./u(:,1)-sqrt(9.81*u(:,1))),abs(u(:,2)./u(:,1)
                            +sqrt(9.81*u(:,1))))','u');
%% Solution of the system with Rusanov
vinit=[hinit,hinit.*uinit];
v=Rusanov(T,L,dx,vinit,f,specrad,2);
plot(space,v(:,1),'k');
hold on;
plot(space,v(:,2),'r');
legend('height','momentum')
```

4. Test it for the compressible Euler system (4) with initial data (5) on the domain $[0,1]$ with $\Delta x = 0.01$. The time step will be adapted at every step according to the stability condition. We will use Neumann boundary conditions.

```
clear;
clf;
%% Space discretization
L=1;
dx=0.01;
space=(0:dx:L)';
%% Time discretization
T=1;
dt=dx*0.95;
%% Initial data
space1=space(space<0.5);
space2=space(space>=0.5);
rhoinit=[ones(size(space1));0.125*ones(size(space2))];
uinit=zeros(size(space));
pinit=[ones(size(space1));0.1*ones(size(space2))];
vinit=[rhoinit,rhoinit.*uinit,(rhoinit.*uinit.^2+pinit)/2];
%% Function and spectral radius of jacobian for the Euler compressible system
%%in the (rho, j,E) variables
f=inline('[u(:,2),2*u(:,3),3*u(:,2).*u(:,3)./u(:,1)-(u(:,2).^3)./(u(:,1).^2)]','u');
specrad=inline('max(abs(u(:,2)./u(:,1)-sqrt(3*(2*u(:,3)./u(:,1)-(u(:,2).^2)./(u(:,1).^2)))
        abs(u(:,2)./u(:,1)+sqrt(3*(2*u(:,3)./u(:,1)-(u(:,2).^2)./(u(:,1).^2)))))','u');
%% Solution of the system with Rusanov
v=Rusanov(T,L,dx,vinit,f,specrad,3);
plot(space,v(:,1),'k');
hold on;
plot(space,v(:,2),'r');
plot(space,v(:,3),'g');
legend('density','momentum','energy')
```