

**AME60714: Advanced Numerical Methods**  
**Homework 2: Due Monday, October 5, 2020**

**Problem 1:** (40 points) Consider a one-dimensional system of nonlinear conservation laws

$$q_{,t} + f(q)_{,x} = 0, \quad q(x, 0) = \overset{\circ}{q}(x), \quad q(a, t) = q_a, \quad q(b, t) = q_b \quad (1)$$

for all  $x \in (a, b)$  and  $t \in (0, T]$ , where  $q(x, t) \in \mathbb{R}^m$  is the conservative state,  $f(q) \in \mathbb{R}^m$  is the (nonlinear) flux function, and  $q_a$  and  $q_b$  are boundary conditions at  $x = a$  and  $x = b$ , respectively. For *periodic* boundary conditions, we define  $q_a = q(b, t)$  and  $q_b = q(a, t)$ . Implement a first-order finite volume method (uniform grid) with explicit fourth-order Runge-Kutta (RK4) timestepping to approximate the solution of (1). For generality and modularity, I suggest implementing two functions: one that advances a general system of ODEs a single time step using RK4 and one that evaluates the first-order finite volume velocity function; see signatures below. By passing the numerical flux function as an argument, the code will be sufficiently general to handle any PDE of the form (1) by simply providing an appropriate numerical flux function.

```

1 function u_np1 = advance_ode_nonlin_rk4(u_n, dt, F)
2 %ADVANCE_ODE_NONLIN_RK4 Advance a nonlinear system of ODEs defined by
3 %
4 %           \dot{u} + f(u) = 0
5 %
6 % from time n to n+1 (step size = dt) using explicit 4th order Runge-Kutta
7 % (RK4).
8 %
9 %Input arguments
10 %-----
11 % U_N : Array (m,) : Solution vector at step n
12 %
13 % DT : number : Time step
14 %
15 % F : function : Function defining the nonlinear velocity vector of ODE
16 %   system (f: Array (m,)). The input to the function is the ODE state
17 %   vector u.
18 %
19 %Output arguments
20 %-----
21 % U_NP1 : Array (m,) : Solution vector at step n+1
    
```

```

1 function F = eval_vel_fvml(Q, dx, numflux_fcn)
2 %EVAL_VEL_FVML Evaluate first-order finite volume velocity given the finite
3 %volume state (Q), uniform element size (dx), and numerical flux function
4 %(numflux_fcn: function of ql and qr)
    
```

For each of the following, plot the solution (all components) at the final time  $t = T$  on a sufficiently refined grid in space and time. Be sure to experiment with the refinement (in space and time) and discuss any relevant observations you make (particularly regarding stability and accuracy of the method).

- a) Solve linear advection with periodic boundary conditions using the exact Riemann solution at  $x/t = 0$  as the numerical flux

$$f(q) = cq, \quad f^*(q_l, q_r) = \begin{cases} q_r & \text{if } c < 0 \\ q_l & \text{if } c > 0 \end{cases}$$

(can be simplified using the known value of  $a$ ). Take  $c = 1$ ,  $b = -a = 1$ ,  $T = 2$ , and the initial condition  $\overset{\circ}{q}(x) = e^{-x^2/0.1}$ . How many elements and timesteps are required to get a reasonable solution? Fix  $\Delta x = 0.02$  and  $\Delta t = 0.002$  and take  $T = 2k$  for  $k = 1, 2, 5, 10$ ; what do you notice about the solution as  $k$  increases? Repeat for  $\Delta x = 0.004$ .

- b) Solve Burgers' equation ( $f(q) = q^2/2$ ) with periodic boundary conditions using the Godunov numerical flux (exact solution of the Riemann problem at  $x/t = 0$ ) in (2). Take  $b = -a = 1$ ,  $T = 0.75$ , and the initial condition  $\overset{\circ}{q}(x) = 0.5 - \sin(\pi x)$ .

c) Solve the Riemann problem for the  $p$ -system

$$q = (v, u), \quad f(q) = (-u, p(v)),$$

with  $p(v) = -e^v$  and the following initial conditions

$$\overset{\circ}{q}(x) = \begin{cases} (1, 3) & \text{if } x < 0 \\ (1, 1) & \text{if } x > 0, \end{cases} \quad \overset{\circ}{q}(x) = \begin{cases} (1, 1.5) & \text{if } x < 0 \\ (2, 1) & \text{if } x > 0, \end{cases} \quad \overset{\circ}{q}(x) = \begin{cases} (2, 1) & \text{if } x < 0 \\ (1, 1.5) & \text{if } x > 0. \end{cases}$$

The boundary states  $q_a, q_b$  can be read from the initial condition. Take  $b = -a = 1$  and  $T = 0.1$ . This corresponds to the same setup from Homework 1, Problem 4d with  $\alpha = 1$ . Use the HLL numerical flux with wave speed estimates from the Roe averages

$$s_1 = -e^{\hat{v}/2}, \quad s_2 = e^{\hat{v}/2}, \quad \hat{v}(q_l, q_r) = \log \frac{e^{v_r} - e^{v_l}}{v_r - v_l}.$$

Include the exact solution on all plots submitted (code that solves the Riemann provided in Homework 1); comment on differences observed between this approach (FVM with numerical flux) and the results obtained from Homework 1, Problem 4d.

As you will be using HLL for the remaining parts of this problem, I suggest writing a function that computes the HLL numerical flux from wave speed estimates.

```
1 function fs = eval_numflux_hll(ql, qr, smin, smax, flux_fcn)
2 %EVAL_NUMFLUX_HLL Evaluate HLL numerical flux from left/right states
3 % (ql, qr), minimum and maximum wave speed estimates (smin, smax), and the
4 % flux function defining the conservation law (flux_fcn).
```

Then you can write a simple wrapper around this function that defines the flux function and wave speed estimates for the  $p$ -system and call `eval_numflux_hll`.

```
1 function fs = eval_numflux_hll_psys(ql, qr)
2 %EVAL_NUMFLUX_HLL_PSYS Evaluate HLL numerical flux for p-system from left
3 % and right states (ql, qr).
```

d) Solve the Riemann problem for the shallow water equations

$$q = (h, hu), \quad f(q) = (hu, hu^2 + gh^2/2),$$

and the following initial condition

$$\overset{\circ}{q}(x) = \begin{cases} (1, -0.5) & \text{if } x < 0 \\ (1.5, 0.75) & \text{if } x > 0. \end{cases}$$

The boundary states  $q_a, q_b$  can be read from the initial condition. Take  $g = 1$ ,  $b = -a = 1$ , and  $T = 0.1$ . Use the HLL numerical flux with wave speed estimates from the Roe averages

$$s_1 = \hat{u} - \sqrt{g\hat{h}}, \quad s_2 = \hat{u} + \sqrt{g\hat{h}}, \quad \hat{h}(q_l, q_r) = \frac{h_l + h_r}{2}, \quad \hat{u}(q_l, q_r) = \frac{\sqrt{h_l}u_l + \sqrt{h_r}u_r}{\sqrt{h_l} + \sqrt{h_r}}.$$

e) Solve the Euler equations (ideal gas law)

$$q = (\rho, \rho v, \rho E), \quad f(q) = (\rho v, \rho v^2 + p(q), [\rho E + p(q)]v),$$

where  $p(q) = (\gamma - 1)(\rho E - \rho v^2/2)$  and the following initial condition

$$\overset{\circ}{q}(x) = (\overset{\circ}{\rho}(x), \overset{\circ}{\rho}(x)v_\infty, \overset{\circ}{p}(x)/(\gamma - 1) + \overset{\circ}{\rho}(x)v_\infty^2/2),$$

where the inflow conditions are  $\rho_\infty = 1$ ,  $v_\infty = 1$ ,  $M_\infty = 0.2$ , and  $p_\infty = \rho_\infty v_\infty^2 / (\gamma M_\infty^2)$ , and the spatial variation is given by

$$\overset{\circ}{\rho}(x) = \rho_\infty(1 + 0.1\Gamma(x)), \quad \overset{\circ}{p}(x) = p_\infty(1 + 0.1\Gamma(x)), \quad \Gamma(x) = e^{-(x-0.3)^2/0.05^2}.$$

The boundary states  $q_a$ ,  $q_b$  can be read from the initial condition. Take  $\gamma = 1.4$ ,  $b = -a = 5$ , and  $T = 0.025$ . Use the HLL numerical flux with wave speed estimates from the Roe averages

$$s_1 = \hat{v} - \hat{c}, \quad s_2 = \hat{v} + \hat{c}, \quad \hat{v} = \frac{\sqrt{\rho_l}v_l + \sqrt{\rho_r}v_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}, \quad \hat{H} = \frac{\sqrt{\rho_l}H_l + \sqrt{\rho_r}H_r}{\sqrt{\rho_l} + \sqrt{\rho_r}},$$

where  $\hat{c} = \sqrt{(\gamma - 1)(\hat{H} - \hat{v}^2/2)}$  and  $H(q) = (\rho E + p(q))/\rho$ .

f) Finally, solve the famous Osher-Shu problem: the Euler equations with the following initial condition

$$\overset{\circ}{q}(x) = \begin{cases} (\rho_l, \rho_l v_l, p_l / (\gamma - 1) + \rho_l v_l^2 / 2) & \text{if } x < -4 \\ (\overset{\circ}{\rho}_r(x), \overset{\circ}{\rho}_r(x)v_r, p_r / (\gamma - 1) + \overset{\circ}{\rho}_r(x)v_r^2 / 2) & \text{if } x > -4, \end{cases}$$

where  $\rho_l = 3.857143$ ,  $v_l = 2.629369$ ,  $p_l = 10.3333$ ,  $v_r = 0$ ,  $p_r = 1$ , and the density disturbance is

$$\overset{\circ}{\rho}_r(x) = 1 + \sin(5x)/5.$$

The boundary states  $q_a$ ,  $q_b$  can be read from the initial condition. Take  $\gamma = 1.4$ ,  $b = -a = 5$ , and  $T = 1$ . Use the HLL numerical flux with wave speed estimates from the Roe averages.

**Problem 2:** (50 points) Implement a high-order discontinuous Galerkin method (uniform grid) with explicit fourth-order Runge-Kutta (RK4) timestepping to approximate the solution of (1). For generality and modularity, I suggest implementing a function that evaluates the DG velocity function (signature below) and combining with the RK4 function from Problem 1 to advance in time. By passing the physical and numerical flux functions as arguments, the code will be sufficiently general to handle any PDE of the form (1) by simply providing an appropriate fluxes.

```

1 function F = eval_vel_dg(Q, T, w, Mi, flux_fcn, numflux_fcn, Qlr)
2 %EVAL_VEL_DG Evaluate high-order discontinuous Galerkin velocity given the
3 %DG state (Q), basis functions (T), quadrature weights (w), element mass
4 %matrix inverses (Mi), physical flux function (flux_fcn: function of q),
5 %and numerical flux function (numflux_fcn: function of ql and qr).
```

The function `setup_dg_msh_disc` provided setups up the basis functions, quadrature weights, and mass matrix inverses.

```

1 function [xdg, wq, T, Mi] = setup_dg_msh_disc(xlims, nelem, porder, nvar)
2 %SETUP_DG_MSH_DISC Setup DG mesh and discretization
3 %
4 %Input arguments
5 %-----
6 % XLIMS : Array (2,) : Domain limits
7 %
8 % NELEM : number : Number of elements
9 %
10 % PORDER : number : Polynomial degree
11 %
12 % NVAR : number : Number of variables in conservation law
13 %
14 %Output arguments
15 %-----
16 % XDG : Array (1, NVPE, NELEM) : Nodal position for each element, i.e.,
17 % XDG(:, :, e) = nodal positions for element e.
```

```

18 %
19 % WQ : Array (NQ,) : Weights for quadrature rule
20 %
21 % T : Array (NDOF_PER_ELEM, NVAR*(NDIM+1), NQ, NELEM) : Basis function and
22 % their derivative for each element evaluated at each quadrature node,
23 % i.e., T(:, 1:NVAR, j, e) = basis functions at quadrature node j of
24 % element e and T(:, NVAR+1:end, j, e) = derivative of basis function at
25 % quadrature node j of element e
26 %
27 % Mi : Array (NDOF_PER_ELEM, NDOF_PER_ELEM, NELEM) : Mass matrix inverse
28 % for each element.
    
```

Use you DG code to repeat Problem 1a-1e (skip 1f since it will not work without stabilization of the shock). For all tests, use 50 elements and consider at least polynomial degrees  $p = 1, 2$ . Comment on the accuracy, stability, dissipation, and cost of the scheme relative to the FVM from Problem 1. How do these observations vary with  $p$ ?

**Problem 3:** (30 points) For a scalar nonlinear conservation law in one-dimension with convex flux function  $f(q)$  ( $f''(q) > 0$ ), prove:

a) the average numerical flux is not an E-flux

$$f_{\text{avg}}^*(q_l, q_r) = \frac{f(q_l) + f(q_r)}{2}$$

b) the Godunov numerical flux is an E-flux

$$f_{\text{godnv}}^*(q_l, q_r) = \begin{cases} \min_{q_l \leq q \leq q_r} f(q) & \text{if } q_l \leq q_r \\ \max_{q_r \leq q \leq q_l} f(q) & \text{if } q_r \leq q_l \end{cases} \quad (2)$$

c) the local Lax-Friedrichs (Rusanov) numerical flux is an E-flux

$$f_{\text{lif}}^*(q_l, q_r) = \frac{f(q_l) + f(q_r)}{2} - \frac{\alpha}{2}(q_r - q_l), \quad \alpha = \max\{|f'(q_l)|, |f'(q_r)|\}$$

(for brevity, assume  $q_r \geq q_l$ ). Hint: since  $f$  is a convex function,  $f'(y)(y - x) \geq f(y) - f(x)$  for any  $x, y$ .

Recall that  $f^*(q_l, q_r)$  is an E-flux corresponding to flux  $f(q)$  if

$$(f(\xi) - f^*(q_l, q_r))(q_r - q_l) \geq 0,$$

for all  $\xi \in [\min\{q_l, q_r\}, \max\{q_l, q_r\}]$ .