# UC Berkeley Math 228B, Spring 2020
# Problem Set 3
# DUE DATE: March 28, 2020

Suncica Canic

March 14, 2020

1. Consider the Burger's equation $u_t + (u^2/2)_x = 0$ in quasilinear form:

$$u_t + u\ u_x = 0, \quad -\infty < x < \infty, t > 0,$$

and the following two Riemann initial data:

**Riemann data I:**

$$u(x,0) = \begin{cases} 0, & x < 0 \\ 1, & x > 0. \end{cases}$$

and

**Riemann data II:**

$$u(x,0) = \begin{cases} 1, & x < 0 \\ 0, & x > 0. \end{cases}$$

(a) Find the exact solution to each Riemann Problem defined above.

(b) Use the following generalization of the classical linear upwind scheme to solve both Riemann problems for the Burger's equation in quasilinear form numerically:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} u_j^n (u_j^n - u_{j-1}^n), \quad x \in (-3,3), t > 0.$$

Use $x \in (-3,3)$ for the computational domain and $u(-3,t) = u_L$ at the left numerical boundary, and $u(3,t) = u_R$ at the right numerical boundary. Here $u_L$ and $u_R$ are the left and right states in the Riemann data. Run the code until $t = 0.8$s.

(c) State the CFL condition that you used.

(d) Study convergence of this method for both Riemann Problems by using the following grid sizes:

$$\Delta x = 6/N, N = 60, 120, 240,$$

using the **numerical solution** obtained with $N = 240$ as the base solution ("exact solution"). Plot the log-log error graph.

(e) For each Riemann Problem state whether the method converges or not.

(f) If the method converges, does it converge to the exact solution?

(g) Compare the exact solution with the numerical solution obtained with $N = 240$ at time $t = 0.8$.

2. Design a conservative upwind method to solve the two Riemann Problems in Problem 1 above.

(a) What is the numerical flux function?

(b) Investigate convergence of your method by applying the same strategy as described in Problem 1(d).

(c) Compare numerical solution to the exact solution for each Riemann Problem for $\Delta x = 6/240$ at time $t = 0.8$s. Superimpose the graphs of the numerical and exact solution versus $x$ at time $t = 0.8$s.

(d) Plot the pointwise difference between the numerical and exact solution at $t = 0.8$.

3. Within the Matlab code(s) provided in this folder, write the main steps of the *Two-Step Lax-Wendroff Method with Strang splitting* to simulate blood flow in a coronary artery under normal, rest conditions, and after administration of adenosine. Adenosine is used in certain patients to check for sufficient coronary reserve to accommodate an increased oxygen demand during exercise. In general, administration of adenosine increases flow resulting from vasodilatation due to adenosine injection, referred to as hyperaemia [1]. Manuscript [1], where details of experiment and an overview of coronary structure and perfusion in health and disease are described, can also be found in the folder. The main focus of this project is on recovering the experimental results in Fig. 4 (top) in [1]. Namely, the main focus is on recovering the blood flow velocity, reported in Fig. 4, using your Matlab solver, and then providing additional information, such as vascular dilatation and flow, not measured during experiment.

After adding the main steps of the two-step Lax-Wendroff method (together with the treatment of boundary data), plot the velocity at the mid-point of the artery as a function of time during one cardiac cycle (the plotting command is already in the code), plot the change in the cross-sectional area at the same mid-point over one cardiac cycle (the plotting command is already in the code), and plot the flow at the mid-point over one cardiac cycle (not provided in the code).

First check that the velocity you calculated using the code is similar to the measured data provided in Fig. 4 (a) and (b) (top) of the manuscript. Then, show, by plotting the flow at the mid-point of the artery, that your solver indeed captures increased flow during hyperaemia. Report what is the maximum flow at the mid-point over one cardiac cycle (second cycle) in both cases: rest and hyperaemia.

Notice that the change in the cross-section area (or radius) of the vessel, is difficult to measure. Thus, your code provides additional information that could not have been captured by "standard" measurements.

Comment on the "strange" results during first cardiac cycle (which is around 0.8 seconds long).

The files you will need are all in Projects/Math228B-Project3-FILES/ folder. They include:

- Two sets of inlet and outlet pressure data, obtained from the measurements in a patient considered in [1], Fig. 4. The files: ProximalPressure_fft.dat and DistalPressure_fft.dat contain the inlet and outlet pressure data at rest. The files: Hyperaemia_ProximalPressure_fft.dat and Hyperaemia_DistalPressure_fft.dat contain the inlet and outlet pressure data corresponding to hyperaemia.

- Two Matlab codes: LaxWendroff_PhysiologicalData_Normal.m and LaxWendroff_PhysiologicalData_Hyperaemia.m. In each code you need to add the main steps of the two-step Lax-Wendroff method. The codes differ in the parameters used. The hyperaemia code has a larger reference radius (corresponding to vasodilatation), and a higher Youngs modulus to account for the nonlinear behavior of the vascular wall tissue (stiffer at larger radii).

- Manuscript [1].

[1] J. Spaan et al. *Coronary structure and perfusion in health and disease, REVIEW* **Phil. Trans. R. Soc. A** (2008) 366, 3137-3153

**Project Submission:** Submit a zip-file on bCourses which contains: a pdf file with answers to the questions in this project, together with the figures showing the results for each problem; and two Matlab files (your can also use another language if you prefer), containing the completed codes.