UC Berkeley Math 228B, Spring 2020
Problem Set 6
DUE DATE: May 15, 2020

Suncica Canic

May 2, 2020

1. Solve the Burger's equation $u_t + \left(\frac{u^2}{2}\right)_x = 0$, for $x \in [-1, 1]$, and $t \leq 0.5$, with initial data $u(x, 0) = sin(\pi x)$, using DG method with the Lax-Friedrichs numerical flux, 1st-order elements (1st-order polynomial basis), and forward Euler time integration.

Use hand calculations to obtain roughly what the solution should look like. Indicate what the exact solution looks like by sketching the following graphs:

- The characteristic curves in the x, t plane (indicate what is the characteristic slope as a function of u);
- The solution u versus x at the following three times: $t = 0$, $t = 0.25$ and $t = 0.5$.

Write the weak formulation of the problem, and calculate by hand the matrix form of the semi-discrete problem with 4 elements:

$$K_1 = [-1, -1/2], K_2 = [-1/2, 0], K_3 = [0, 1/2], K_4 = [1/2, 1]$$

and linear basis functions.

Write, by hand, what the semi-discrete problem in matrix form looks like.

2. The Matlab code `BurgersLinearStudentVersion.m`, placed in the folder Project6/Burgers/ on bCourses, contains the main steps in solving the Burger's equation problem, discussed above, using a DG approach. The code is based on implementing linear polynomials with an *arbitrary* number of elements K in the "triangulation" T_h. The only part missing is the implementation of the Lax-Friedrichs numerical flux in the function `[rhs] = BurgersRHS(u)`. Implement the Lax-Friedrichs numerical flux and then run the code with $N = 4, 10, 20$ and 50 elements. Plot the solutions at $t = 0, t = 0.25$ and $t = 0.5$. Comment on what you observe for different values of elements N in the triangulation T_h.

3. The Matlab code `BurgersStudentVersion.m`, placed in the folder Project6/Burgers/ on bCourses, contains the main steps in solving the Burger's equation problem, discussed above, using a DG approach involving *Lagrange polynomials basis functions*, expanded in terms of the orthogonal Legandre basis (as described in "Additional DG notes" in our last lecture). The problem is solved in "strong form" (integrated by parts twice, as described in "Additional DG notes"), and Legendre-Gauss-Lobatto quadrature points are used as the grid nodes within each element. The Vandermonde matrix V is used to help calculate the inverse of the mass matrix M: $M^{-1} = VV^T$ (as described in "Additional DG notes"), and a `minmod` flux limiter is used to limit the oscillations near shocks.

(a) Run this code with $P = 3$ corresponding to a 3rd-order polynomial approximation, on the grid with $N = 20$ elements. Plot u versus x at times $t = 0.25$ and $t = 0.5$.

(b) Run the code obtained in Problem 2 corresponding to using 1st-order polynomials, with $N = 20$ elements. Plot u versus x at times $t = 0.25$ and $t = 0.5$.

(c) Finally, run the code obtained in Problem 2 corresponding to using 1st-order polynomials, with $N = 50$ elements. Plot u versus x at times $t = 0.25$ and $t = 0.5$.

Compare the results from (a) with the results from (b) and (c). Superimpose the plots for $t = 0.25$ using different colors for u corresponding to different approaches. Do the same for $t = 0.5$.

What can you observe? How does the polynomial degree relate to the number of elements for a given accuracy?

Project Submission: Submit your pdf file with the solutions on bCourses.