

Homework 7
due Thursday, Nov. 29

Work in groups of 2-3. Turn in one assignment per group.
Partial credit will be given, no re-grades.

1. (3 points) Write a spectral code (Fourier collocation, Lec. 23) to solve Burgers' equation

$$\begin{aligned} u_t &= -uu_x + \nu u_{xx}, & (\nu = 0.01), \\ u(x, 0) &= \sin x, & (0 \leq x \leq 2\pi), \\ u(0, t) &= u(2\pi, t) & (t > 0) \end{aligned}$$

using the six stage, fourth order IMEX Runge-Kutta scheme (ARK4) at the end of Lecture 22 (the coefficients of the scheme are posted on the course webpage). Discretize space with N equally spaced points $x_j = 2\pi j/N$, $0 \leq j < N$. Use the embedded formula of the scheme to implement stepsize control (see Lectures 12 and 16) with tolerances $\text{Atol} = 1.0 \times 10^{-10}$, $\text{Rtol} = 0.0$. For several choices of N (e.g. 128, 512 and 2048), plot the solutions $u(x, T_i)$ with $T_i = 3i/32$, $0 \leq i \leq 32$. (Plot them all on the same graph, u vs. x). Also plot the magnitude of the first $N/2$ Fourier modes $|\hat{u}_k(T_i)|$. (Plot them all on the same graph $|\hat{u}_k|$ vs. k , $0 \leq k \leq N/2$). The magnitudes of the highest frequency Fourier modes are a good indication of whether N is large enough to resolve the solution in space. Note: due to stepsize control, your solution is not likely to land exactly at the given times T_i ; use Hermite interpolation (Lecture 16) to interpolate between the times t_n and t_{n+1} that step over each T_i . Finally, make a plot of the stepsizes used by your algorithm for the $N = 2048$ case.

2. (3 points) Now forget about stepsize control and freeze $N = 1024$. Do a convergence study of the solution at time $T = 1.0$ by running your code for several values of $h = \Delta t$ and making a log-log plot of the error vs. h . (Use the result from the smallest timestep for the exact solution and define error = $\sqrt{\frac{1}{N} \sum_{j=0}^{N-1} |u_j(T) - u(2\pi j/N, T)|^2}$). I found the following values of h to work well:

$$h = [1, 2, 3, 4, 6, 8, 12, 16]/8192$$

Code up the second and fourth order IMEX multistep schemes (SBDF2 and SBDF4 in Lecture 21) and repeat the convergence study calculation for these algorithms. (Use the Runge-Kutta solution to start the multistep methods going). I found SBDF4 was slightly unstable when $h=16/8192$ in the above list, and was corrupted by roundoff error when $h=1/8192$, so throw away those two data points when computing the convergence rate.

3. (3 points) repeat 1 for the KdV equation:

$$u_t = -uu_x - \nu u_{xxx}, \quad (\nu = 0.01).$$

4. (2 points) repeat 2 for the KdV equation for the ARK4 and SBDF2 schemes.

5. (1 point) The lack of A-stability of the 4th order BDF method makes SBDF4 perform poorly on the KdV equation. Illustrate this by running SBDF4 with $N = 256$, $h = 1/60000$ to time $T = 1$; plot the resulting $u(x, T_i)$ and $|\hat{u}_k(T_i)|$ for $T_i = i/32$, $0 \leq i \leq 32$. (Larger values of h or N lead rapidly to inf's and NaN's).