Università
della
Svizzera
italiana

Faculty
of Informatics

**Institute of
Computational
Science
ICS**

Advanced Discretization Methods
Spring Semester 2020

Prof. Igor Pivkin

Exercise Sheet 2

Date of submission: 23.04.2020

Exercise 1

Repeat exercise 3 from homework 1 using the even method. Compare your results with the results you previously obtained using the odd method. Which method is more accurate?

Exercise 2

This exercise is based on the example in the book.. but it will be useful for the exam project.
Consider scalar hyperbolic problem given as

$$\frac{\partial u(x,t)}{\partial t} = -2\pi \frac{\partial u(x,t)}{\partial x},$$
$$u(0,t) = u(2\pi, t),$$
$$u(x,0) = \exp[\sin(x)],$$

where $u(x,t) \in C^\infty[0, 2\pi]$ is assumed periodic and the initial condition is assummed periodically extended.
The problem has an analytic solution given as

$$u(x,t) = \exp[\sin(x - 2\pi t)].$$

This solution represents the initial condition which propagates at the velocity 2π towards increasing x.
Construct a program that solves this problem using an equidistant grid given as

$$x_j = \frac{2\pi}{N+1} j = j\Delta x, \qquad j \in [0, N],$$

and use the exact solution at $t = 0$ as the initial condition.
To advance the equation in time, use 4th order Runge-Kutta method obtained by first defining the function, $F(u^n)$, as

$$\frac{du^n}{dt} = F(u^n) = -2\pi \frac{\partial u^n}{\partial x},$$

where u^n represents the solution at $t = n\Delta t$ and Δt is the time step. The 4th order Runge-Kutta method for advancing from u^n to u^{n+1} is then defined as

$$u_1 = u^n + \frac{\Delta t}{2} F(u^n)$$

$$u_2 = u^n + \frac{\Delta t}{2} F(u_1)$$

$$u_3 = u^n + \Delta t F(u_2)$$

$$u^{n+1} = \frac{1}{3}\left(-u^n + u_1 + 2u_2 + u_3 + \frac{\Delta t}{2} F(u_3)\right),$$

where u_1, u_2 and u_3 are introduced for convenience as help-functions/arrays.
Use sufficiently small time step, Δt, to ensure stability and to reduce time-stepping errors.
Use three different approximations of the spatial derivative:

- The local second order centered finite difference approximation to the spatial derivative of $u(x,t)$ at x_j

$$\frac{du}{dx}\,|_{x_j} = \frac{u_{j+1} - u_{j-1}}{2\Delta x}.$$

- The local forth order centered finite difference approximation

$$\frac{du}{dx}\,|_{x_j} = \frac{u_{j-2} - 8u_{j-1} + 8u_{j+1} - u_{j+2}}{12\Delta x}.$$

- The global infinite order approximation using Fourier differentiation matrix

$$\frac{du}{dx}\,|_{x_j} = \sum_{i=0}^{N} \tilde{D}_{ji} u_i,$$

  where the entries of the matrix operator are

$$\tilde{D}_{ji} = \begin{cases} \frac{(-1)^{j+i}}{2}\left[\sin\left(\frac{(j-i)\pi}{N+1}\right)\right]^{-1} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}.$$

(a) Measure the L^∞-error between the computed solution using three approximatins of spatial derivative and the exact solution at time $t = \pi$ for $N = 8, 16, 32, 64, 128, 256, 512, 1024, 2048$. What are the convergence rates observed and do they correspond to what you would expect? If you measure error for $N = 2048$ using second order scheme, what N do you need to obtain the same error using fourth order and infinite order schemes?

(b) Compare the performance of second order and infinite order schemes for long time integration. For both schemes, plot the computed and the exact solutions at time $t = 0, 100, 200$. For second order scheme use $N = 200$ and for infinite order scheme use $N = 10$.