AOS 180 – Modeling Task 3

A linear wave equation – part II

Due date: April 26$^{\text{th}}$11:00AM

Problem description

The problem of interest is the same as in the Modeling Task 2, with the same set of basic parameters (except that now we will increase the total integration time to $T_{tot} = 2000\,\mathrm{s}$, corresponding to 20 complete cycles for the analytical solution with wavelength $\lambda_x = 50\,\mathrm{m}$). We will test different discretization approaches here, and try to document the features of these methods.

Numerical approach

We will first use the leapfrog scheme with centered second-order differencing in space (the one we discussed in class) to solve this problem. Then we will use the multi-stage third-order Runge-Kutta (RK3) method together with the same centered second-order differencing in space. We will also produce some runs with the model from Modeling Task 2 for comparison. We will also use the same metrics of Modeling Task 2 (i.e. the amplitude of the wave and the discretization error) to assess the different models.

The main goal of our numerical experiments is to compare all three discretization methods (FTBS, leapfrog, and RK3) in the simple wave problem and assess how their performance changes as we change the Courant number (by changing Δt) and the wavelength (actually $\lambda/\Delta x$). In the table below you will find the set of numerical experiments that I suggest you conduct. Of course, feel free to design a number of additional tests to get a better understanding of the behavior of your model.

Exp.	Δt	λ_x	Main goal
#1	$2\,\mathrm{s}$	$50\,\mathrm{m}$	Base case for code testing with optimal C
#2	$\neq 2\,\mathrm{s}$	$50\,\mathrm{m}$	Assessing effects of changing C (use at least 2 cases with $\Delta t = 1, 4\,\mathrm{s}$)
#3	$2\,\mathrm{s}$	$10\,\mathrm{m}$	Repeat experiment #1 using smaller wavelengths
#4	$1\,\mathrm{s}$	$25\,\mathrm{m}$	Solution for a square wave

Report

Please include in your report:

i. Figures: Since this is very similar to Modeling Task 2, no additional information is needed here.

ii. Documentation: Please include in your report the recursive equations used and how you implemented the boundary conditions. Also include a table of your experiments containing the basic configuration for each run together with C and $\lambda_x/\Delta x$. Also include other things you think may be useful.

iii. Discussion: Please briefly discuss each one of your numerical experiments in your report. A few sentences per experiment is enough. In particular, I would like you to document and discuss the amplitude and phase errors for all schemes and the presence/absence of the computational mode in the leapfrog scheme. Some simulations may have interesting features in the behavior of $\epsilon_u(t)$ – can you explain? The goal of your work should be to document and point out advantages and disadvantages of each one of the 3 methods.

Please upload a **.pdf** file with your report and a file with your code to the dropbox that will be available on the CCLE website for the course.

Additional suggestions

You are not required to do these, and even if you decide to do so, you are not required to hand them in (but you can if you want to). Keep this in mind: improving your code now will result in a better code at the end, which will allow you to perform better simulations for your final project. Getting some experience with AB3 and a higher-order scheme in space may be a good idea...

- Can you find out how much you can push C before the RK3 becomes unstable?
- My favorite time advancement scheme is the Adams-Bashforth 3 (AB3). If you have some time, go ahead ant try it on this problem. It is quite easy to modify your leapfrog code to make it into an AB3 time discretization. The discretization in time is given by

$$\frac{u^{n+1}-u^n}{\Delta t} = \frac{23}{12} f\left(u^n\right) - \frac{16}{12} f\left(u^{n-1}\right) + \frac{5}{12} f\left(u^{n-2}\right), \tag{1}$$

  and the recursive equation is

$$u_i^{n+1} = u_i^n - \frac{1}{24} C\left[23\left(u_{i+1}^n - u_{i-1}^n\right) - 16\left(u_{i+1}^{n-1} - u_{i-1}^{n-1}\right) + 5\left(u_{i+1}^{n-2} - u_{i-1}^{n-2}\right)\right]. \tag{2}$$

- If you still want more fun, you can try a higher order differencing in space with leapfrog, RK3, and/or AB3. For example, for the leapfrog scheme you can use

$$u_i^{n+1} = u_i^{n-1} - \frac{1}{6} C\left(-u_{i+2}^n + 8u_{i+1}^n - 8u_{i-1}^n + u_{i-2}^n\right). \tag{3}$$

  What are the practical challenges associated with a higher order spatial scheme?