

Second Programming Assignment, MECH510

Shayan Hoshiyari
Student #: 81382153

November 2015

Contents

1	The Problem	1
2	Numerical Method	1
2.1	Space Discretization Scheme	2
2.2	Explicit Euler	2
2.3	Second Order Runge-Kutta	2
2.4	Two Step Adams-Bashforth	3
2.5	The Final Touches	3
3	Results	4
3.1	Verification	5
3.2	Stability	6
3.3	Time Advance Scheme	7
4	Conclusion	9

1 The Problem

In this assignment the linear wave equation, Equation (1), will be solved numerically.

$$T_t + uT_x = 0 \tag{1}$$

$$\text{with } T(x, 0) = f(x) \quad T(0, t) = g(t)$$

In this equation T is the unknown variable and u is a constant velocity. This equation has an analytic solution in the form of Equation (2), which will be used to assess the accuracy of our numerical methods.

$$T(x, t) = \begin{cases} g(t - \frac{x}{u}) & x < ut \\ f(x - ut) & x > ut \end{cases} \tag{2}$$

2 Numerical Method

Solving the linear wave equation has two main parts: space discretization and time discretization. In this section we will explain each part briefly.

2.1 Space Discretization Scheme

To discretize the equation in space we use the finite volume method and the second order upwind interpolation scheme to transform the original PDE into a set of algebraic differential equations. This system is shown in Equation (3). Note that in this report cell numbering starts from one, while no ghost cells are used. Also N_c denotes the number of cells.

$$\frac{d\bar{T}_i}{dt} = \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} \quad i = 1, 2, \dots, N_c \quad (3)$$

$$F_{i+1/2} = \begin{cases} -ug(t) & i = 0 \\ -u(2T_1 - g(t)) & i = 1 \\ -u(3T_i - T_{i-1})/2 & i = 2, 3, \dots, N_c \end{cases}$$

The second part of the numerical method is time integration, i.e., the system of ODE's (3) should be solved using an appropriate time advancing scheme. To understand this process better, we first write Equation (3) in vector form.

$$\frac{d\mathbf{T}}{dt} = \mathbf{F}(\mathbf{T}, t)$$

Note that at any time t_n given the control volume average values \mathbf{T} , $\mathbf{F}(\mathbf{T}, t)$ can be calculated using equation (3). In this assignment three different time advancing schemes will be considered: second order Runge-Kutta (RK2), second order Adams-Bashforth(AB2), and Explicit Euler.

2.2 Explicit Euler

The Explicit Euler scheme is quite simple and can be written in the form of Equation (4). Note that in all the equations X_n means the value of X at time step n .

$$\mathbf{T}_{n+1} = \mathbf{T}_n + \Delta t \mathbf{F}(\mathbf{T}_n, t_n) \quad (4)$$

Not only is the Explicit Euler Scheme first order accurate in time, but it also has stability issues. Figure 1 shows the stability region of the Explicit Euler Scheme and the eigenvalues times CFL number ($\lambda \cdot \text{CFL}$) for the wave equation discretized using the second order upwind scheme. As you can see even for CFL number 0.15 there are still some unstable eigenvalues near the origin. In this assignment the Explicit Euler Scheme will only be used as a starter for the AB2 scheme.

2.3 Second Order Runge-Kutta

The RK2 scheme that we will use in this assignment is shown in Equation (5). Although this is not the only RK2 scheme out there, it can be shown that all RK2 schemes produce the exact same result when applied to linear system of ODE's.

$$\mathbf{T}_* = \mathbf{T}_n + \frac{\Delta t}{2} \mathbf{F}(\mathbf{T}_n, t_n)$$

$$\mathbf{T}_{n+1} = \mathbf{T}_n + \Delta t \mathbf{F}(\mathbf{T}_*, t_n + \frac{\Delta t}{2}) \quad (5)$$

The RK2 scheme is second order, and according to Figure 2 it will be stable for $\text{CFL} < 0.5$.

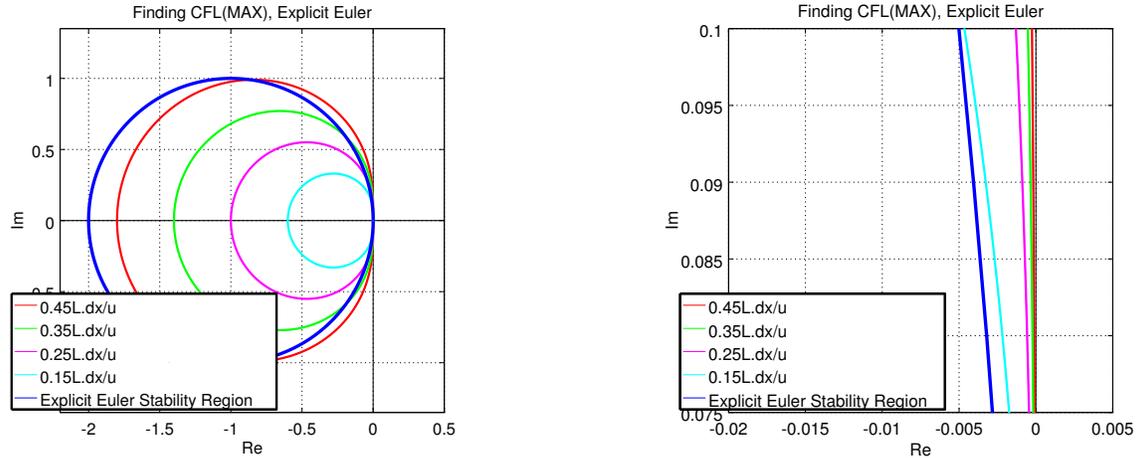


Figure 1: Stability analysis for Explicit Euler. The locus of $\lambda \cdot \text{CFL}$ and the stability region of Explicit Euler are shown in the left figure. While you might think that a CFL number of 0.15 might make the scheme stable, the zoomed right figure shows that even then there are still some eigenvalues outside the stability region.

2.4 Two Step Adams-Bashforth

The Adams-Bashforth Schemes are considered as multi-step schemes, i.e., they use data from previous time-steps to find the value of solution at the new time step. The AB2 method is shown in Equation (6).

$$\mathbf{T}_{n+1} = \mathbf{T}_n + \frac{\Delta t}{2} (3\mathbf{F}(\mathbf{T}_n, t_n) - \mathbf{F}(\mathbf{T}_{n-1}, t_{n-1})) \quad (6)$$

A challenge using these schemes is that they are not self starting. This means that the value of solution at some (in our case two) beginning time-steps must be known if we want to use these schemes. To overcome this problem we consider three solutions:

1. Using of the exact solution for the first time step. Of course this solution applies only to test problems.
2. Using the unstable, lower order Explicit Euler to find the solution at the first time step. We can use smaller time steps at this stage to alleviate the low accuracy of the Explicit Euler Scheme.
3. Using RK2 to find the solution at the first time step.

Finding the stability region of the AB2 scheme is a little more challenging than single-step methods like RK2, and is the subject of our third homework assignment. Here, we will only use the results from [Tea]. In Figure 2 the stability region of AB2 and the eigenvalues of the wave equation discretized using the second order upwind method are shown. It is clear that the scheme is stable for CFL numbers below 0.25.

2.5 The Final Touches

Now that we have explained our space and time discretization schemes our numerical method is almost ready. However, there are a few remaining remarks.

Firstly, the final simulation time must be an integer multiplied by the time step. This means that we can not choose arbitrary CFL numbers. In reality, the code gets a CFL number from the

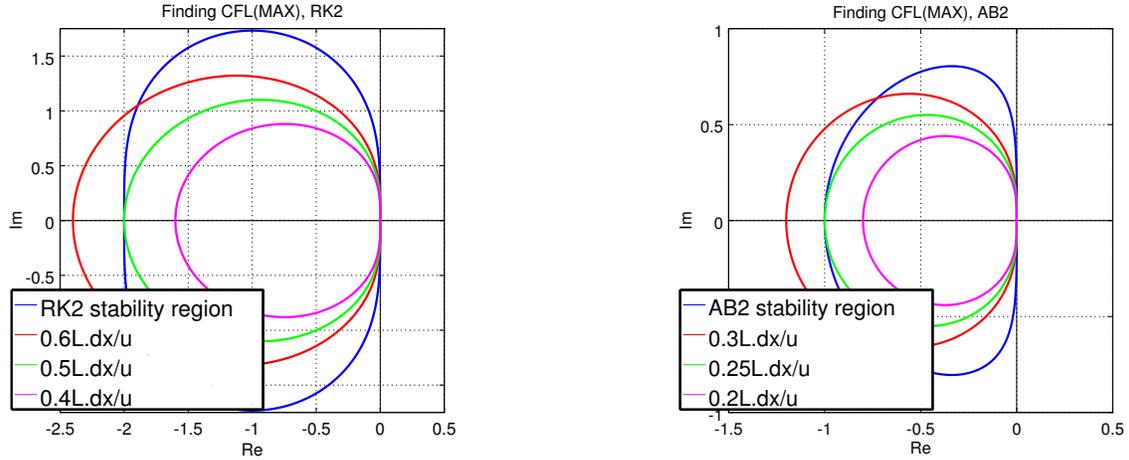


Figure 2: Stability analysis for RK2 and AB2. The locus of $\lambda \cdot \text{CFL}$ and the stability regions of RK2 and AB2 are shown on the left and right respectively. It can be seen that RK2 is stable for $\text{CFL} < 0.5$ while AB2 is stable for $\text{CFL} < 0.25$. On the other hand, RK2 needs two flux evaluations per time step while AB2 requires one, so it seems that the computational time will be almost the same.

user as input, then it chooses a time step which can produce a CFL number as close to the one given as input as possible, which in turn creates a slightly different actual CFL number.

Finally, we summarize the solution procedure in the form of Algorithm 1. With our numerical method fully explained, we will start tackling the problems in the next section.

Algorithm 1 Numerical Solution of the linear wave equation

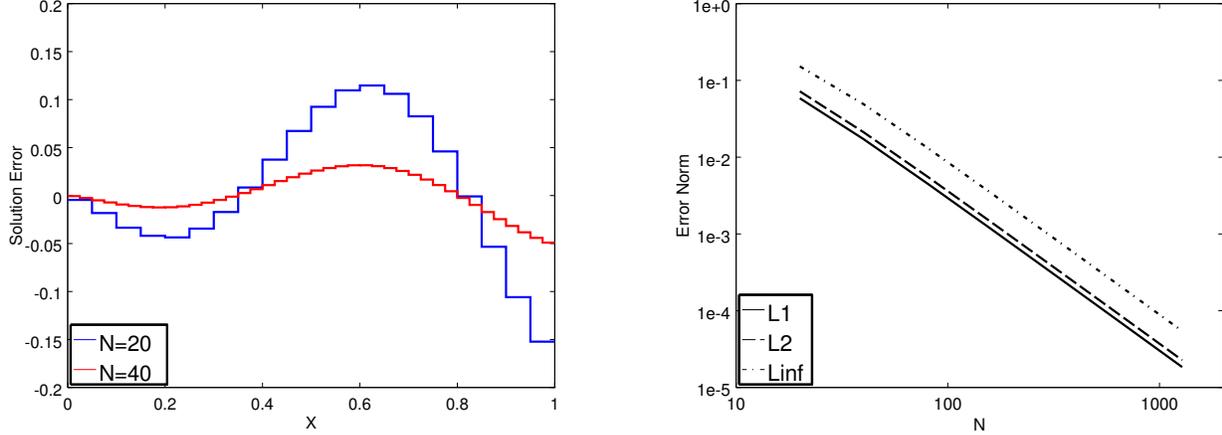
Require: $N_c, x_{max}, u, f(x), g(t), t_e$ final solution time, time advance method

Ensure: \mathbf{T} at t_e , and $\|\mathbf{T}\|_*(t)$

- 1: Find actual Δt and CFL number.
 - 2: Find the average values of T at $t = 0$, i.e., \mathbf{T}_0 using midpoint integration rule.
 - 3: $n = 0$
 - 4: **if** method is AB2 **then**
 - 5: Find \mathbf{T}_1 using either: exact solution and midpoint integration rule, RK2, or Explicit Euler.
 - 6: $n = n + 1$
 - 7: **end if**
 - 8: **while** $n < t_e/\Delta t$ **do**
 - 9: Find \mathbf{T}_{n+1} using appropriate time advance scheme.
 - 10: $n = n + 1$
 - 11: Find solution error norms using the exact solution, and midpoint integration rule.
 - 12: Print $n\Delta t, L_1, L_2$ and L_∞ norms of error.
 - 13: **end while**
 - 14: Find solution error and print it for each CV.
-

3 Results

In this section the solutions to each parts of the assignment are presented.



(a) Solution error at each CV for 20 and 40 control volumes. (b) Behavior of error norms with respect to mesh refinement.

Figure 3: Error plots for the verification problem.

3.1 Verification

In this part we will set $u = 2$, $x_m = 1$, $t_e = 1$, $f(x) = -\sin(2\pi x)$, and $g(t) = \sin(4\pi t)$. With a safety factor of 80%, the input CFL number will be set to 0.4. Using these values as input, the exact solution will reduce to:

$$T(x, t) = \sin(2\pi(2t - x))$$

Firstly, we will solve the problem for $N_c = 20$ and $N_c = 40$. The error plots at $t_e = 1$ are shown in Figure 3a. With the reduction in error by mesh refinement and the lack of spurious oscillations in the solution, we are pretty happy with our implemented scheme.

To study the solution behavior further, the error norms will be calculated for different mesh sizes (20, 40, 80, 160, 320, 640, and 1280). The results are shown in Figure 3b. The slopes of these curves, using the least square method, are -1.949 , -1.949 and -1.926 for the L_1 , L_2 and L_∞ respectively, showing that our scheme is truly second order for the linear wave equation with sine initial condition.

To find the mesh sizes that drop the L_2 norm of the error below a certain value, we use the equation for $L_2(N_c)$. According to Figure 3b:

$$\log L_2 = -1.9491 \log N_c + 1.4330$$

If we want the L_2 error norm to drop below 10^{-3} , N_c must be:

$$\log(10^{-3}) \geq -1.9491 \log N_c + 1.4330 \Rightarrow N_c \geq 189$$

If we want the L_2 error norm to drop below 10^{-4} , N_c must be:

$$\log(10^{-4}) \geq -1.9491 \log N_c + 1.4330 \Rightarrow N_c \geq 613$$

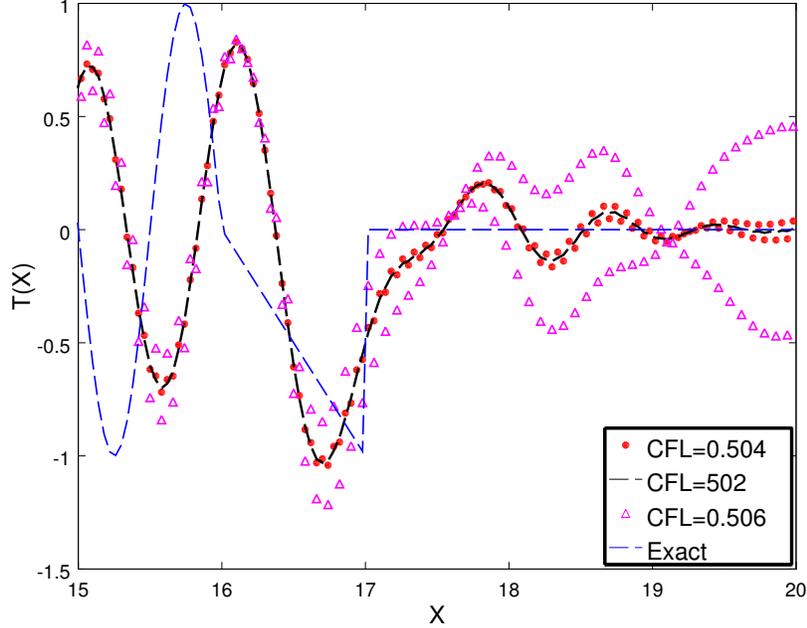


Figure 4: Solution of the stability problem for different CFL numbers. Due to the time step restriction ($\frac{t_e}{\Delta t} = N_t \in \mathbb{N}$) any CFL number can not be chosen and the displayed solutions were actually obtained with the CFL's 0.501882, 0.503778, 0.505689. For CFL= 0.502 the solution is stable, CFL= 0.504 is on the borderline and CFL= 0.506 is slightly unstable.

3.2 Stability

In this part we will set $u = 2$, $x_m = 20$, $t_e = 8$, $N_c = 500$, and $g(t) = \sin(4\pi t)$ while the initial condition is set to:

$$f(x) = \begin{cases} -x & 0 \leq x \leq 1 \\ 0 & \text{elsewhere} \end{cases}$$

To experimentally identify the maximum stable time step, we will run the code for different CFL numbers, starting with 0.490 and increasing the time step by 0.001 each time. My choices for the stable, borderline and slightly unstable cases are CFL's 0.502, 0.504 and 0.506 respectively, suggesting that choosing CFL= 0.502 as the maximum stable CFL with an accuracy of $\pm 0.003\text{CFL}$ is a justified choice. The corresponding Δt and N_t for this CFL are 0.010038 and 797 respectively. The eigenvalue stability analysis presented in Subsection 2.3 suggests CFL= 0.5 as the maximum stable CFL which is consistent with the results obtained in this section.

Before proceeding to the next problem, I must provide my criterion for identifying a stable time step. Our numerical method is not monotonic, so all the solutions will present undershoots and overshoots. However, when CFL=0.504, the numerical solution becomes oscillatory with a wavelength less than the cell size. As the amplitude of these oscillations are small relative to the previously present overshoots and undershoots, I will consider this case as the borderline. When CFL=0.506 the amplitude of low wavelength oscillations becomes as big as the amplitude of the initial wave, so this case is clearly unstable. If we further increase the CFL to 0.507 the solution will blow up (not shown in Figure 4).

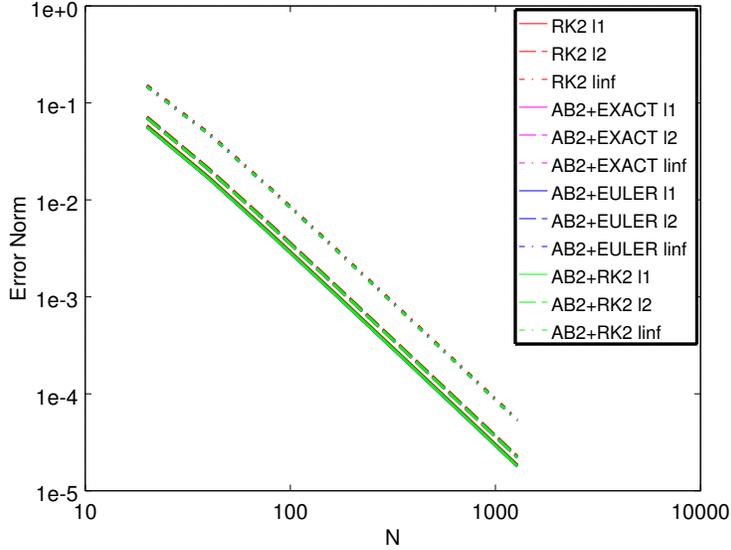


Figure 5: Error norms for different time marching schemes with respect to mesh refinement.

3.3 Time Advance Scheme

In this part we will set all the input to that of Subsection 3.1. However, AB2 will be chosen as the time advance scheme. Using a safety factor of 0.8 The CFL number will be 0.2.

We will use the three mentioned methods in Subsection 2.4 to find the solution at $t = \Delta t$, and then compare our hybrid AB2 schemes with the RK2 scheme.

In Figure 6 the error norms are shown as a function of time. The AB2 schemes are almost identical except for a very slight difference in the beginning. As we expect the error in AB2+EXACT is the least and the error in AB2+EULER is the most. The error in RK2 is slightly more than the error in AB2 and it can be due to the fact that the time step of the latter is half of the former due to stability restrictions.

In Figure 5 the error norms at $t = 1$ are shown for each scheme. The behavior of these schemes are truly unrecognizable, by looking at this graph. The slopes of this graph also confirm that our methods are second order.

The computational cost of RK2 and AB2 are almost the same because the former needs two flux evaluations per time step while the latter needs one flux evaluation with a smaller time step. Although in this example the choice of time advance scheme did not have a significant effect on the solution accuracy, it might not be the case for nonlinear system of equations which can also have discontinuities like shocks. The time advance scheme should be chosen wisely to overcome the following tasks:

- Does not have a heavy stability restriction over Δt .
- Require low computational cost.
- Have the desired order of accuracy.
- Have the desired damping effect, as we might require certain waves with certain frequencies to be damped while others to be magnified.
- ...

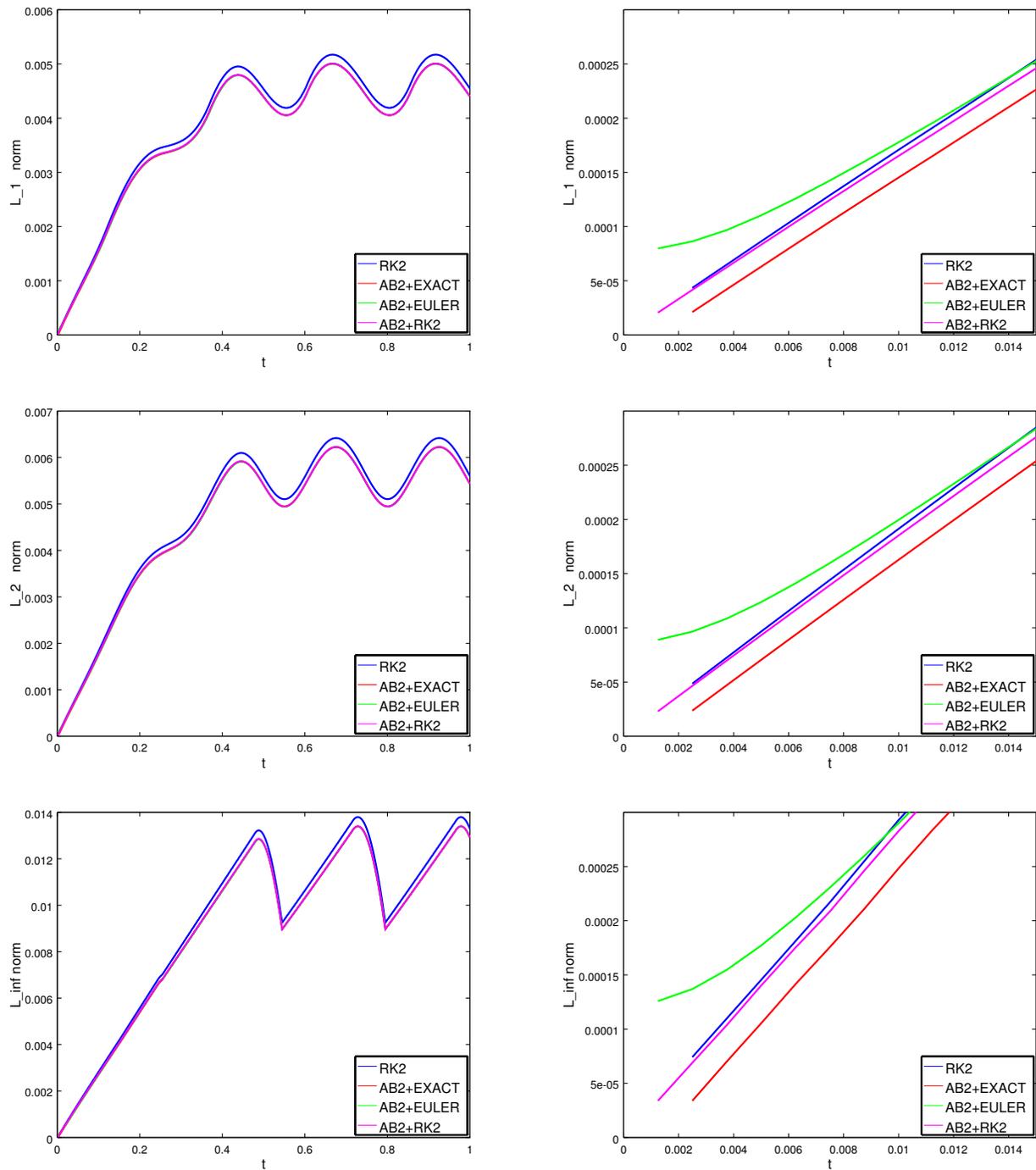


Figure 6: Error norms for the third problem as a function of time. From top to bottom: L_1 , L_2 and L_∞ norms. On the left the results for the whole time are shown. However, as the difference between the AB2 schemes can not be seen, zoomed plots are shown on the right.

4 Conclusion

In this assignment the linear wave equation was solved using the second order upwind scheme and AB2 and RK2 time advance schemes. The error norms were plotted for different mesh sizes and it was verified that the methods are second order. It was also shown that the starting method for AB2 does not have a significant impact on its performance. We also found the maximum stable CFL for RK2 scheme experimentally which was in good agreement with the eigenvalue stability analysis.

It is worth mentioning that despite their great performance for continuous initial conditions, our methods do not perform well when discontinuities are present in the initial condition.

References

[Tea] Chebfun Team. Stability regions of ode formulas. <http://www.mathworks.com/matlabcentral/fileexchange/23972-chebfun/content/chebfun/examples/ode/html/Regions.html>. Accessed: 2015-11-08.