$$|I(x,t)| \sim \frac{1}{\sqrt{t}}|H(\lambda)|. \tag{798}$$

The amplitude of the plane wave will decay as $\frac{1}{\sqrt{t}}$. Or to put it another way; if we shift to a moving frame moving at speed $\omega'(\lambda)$ with respect to the lab frame we would see a plane wave whose amplitude decays as $\frac{1}{\sqrt{t}}$ and whose frequency $\omega(\lambda)$ and wavelength $\frac{2\pi}{\lambda}$ will depend on our speed. Observe that $\omega'(\lambda)$ is just the group velocity that we introduced in (769).

One can prove that for the Klein-Gordon equation, and also for many other dispersive equations, the energy in the wave field is transported at the group velocity.

The group velocity is the relevant velocity for dispersive wave equations. It is also a concept whose importance for the description of wave phenomena in general hardly can be overstated.

12 Projects

12.1 Project1

In this project the aim is to develop and implement on a computer a finite difference algoritm for the solution of the 1D wave equation. Deriving the finite difference algoritm and implementing it in a computer code is not really the challenging and time consuming step here. The real challenge is to make a convincing case for the correctness of the computer implementation. As I have stressed in my lectures we should never, ever, trust anything that comes out of a computer without extensive testing of the implemented code.

So how do we proceed to build the needed trust in our implementation? The best way to do this is to run the code in a situation where an analytic solution is known. Most of the time an analytic solution also needs to be implemented numerically, it typically include infinite sums of functions and integrals that can not be solved in terms of elementary functions. So what we are really doing is to compare the output of two different computer implementations for the same solution. If this is to make sense and build confidence, it is of paramount importance that the two codes implementing the finite difference algoritm and the analytical solution should be as independent as humanly possible. Under such circumstances closely similar output from the two codes will build confidence that our finite difference solution is correct.

So the challenge then is to find analytical solutions to the problem at hand, in our particular case this is the 1D wave equation subject to some boundary conditions that we will specify shortly. It is typically the case that we will not be able to find analytic solutions to the actual problem we are trying to solve but to a related problem or a special case of the problem where the domain, boundary conditions and/or equation is changed somewhat or fixed by some specific choises. After all, if we could find an analytic solution to the actual problem it would not be much point in spending a lot of effort in trying to

solve it using finite differences. It is however important that changing from the actual problem to the related problem does not involve major modifications to the core of the finite difference code. The reason for this is that such major modifications are their own source of error and as a consequence our test will not be very convincing as a test of the actual problem.

So what is the actual problem then? Here it is:

$$\begin{aligned} u_{tt} - u_{xx} &= 0 \quad -l < x < l \\ u(-l,t) &= h(t) \\ u(l,t) &= k(t) \\ u(x,0) &= \varphi(x) \\ u_t(x,0) &= \psi(x) \end{aligned}$$

a) Implement a finite difference method for this problem. The code should be flexible enough to handle arbitrary choises for the functions h,k ,φ and ψ.

b) Show that

$$u(x,t) = \frac{1}{2}(\varphi(x+t) + \varphi(x-t))$$

is a solution to the problem

$$\begin{aligned} u_{tt} - u_{xx} &= 0 \quad -\infty < x < \infty \\ u(x,0) &= \varphi(x) \\ u_t(x,0) &= 0 \end{aligned}$$

In your finite difference code choose l and φ in such a way that the interval is much larger than the support of φ.(By the support of φ I mean the set of points where the function is nonzero). A Gaussian function of the form

$$\varphi(x) = ae^{-b(x-x_0)^2}$$

is a good choise if the parameters a, b and x_0 are choosen in the right way. Also choose $h = k = 0$. As long as the support of the solution u is well away from the boundary of the interval $[0, l]$ our exact solution and the finite difference solution should match. This is our first test.

c) Our second test involve a rather general idea called an "artificial source". We consider the modified problem

$$\begin{aligned} u_{tt} - u_{xx} &= \rho(x,t) \quad -l < x < l \\ u(-l,t) &= h(t) \\ u(l,t) &= k(t) \\ u(x,0) &= \varphi(x) \\ u_t(x,0) &= \psi(x) \end{aligned}$$

This equation would appear as the evolution equation for a string clamped at both ends and where there is a precribed force acting along the string. This

kind of prescribed influence is called a "source". In the artificial source test we turn the problem around and rather assume a solution of a particular type and then compute what the corresponding source must be.

We thus pick a function $u(x,t)$ and calculate the source $\rho(x,t)$, boundary functions $h(t), k(t)$ and initial data $\varphi(x)$ and $\psi(x)$ so that $u(x,t)$ is in fact a solution to our modified problem. Thus

$$\begin{aligned} \rho(x,t) &= u_{tt}(x,t) - u_{xx}(x,t) \\ \varphi(x) &= u(x,0) \\ \psi(x) &= u_t(x,0) \\ h(t) &= u(-l,t) \\ k(t) &= u(l,t) \end{aligned}$$

We now have a analytic solution to our modified problem and can compare this solution to the solution produced by our finite difference code. You are free to choose the function $u(x,t)$ for yourself but here is a possible family of choises

$$u(x,t) = ae^{-b(x-x_0\cos(\omega t))^2}$$

This is a family of Gaussian functions whose center move back and forth in a periodic manner determined by the parameters x_0 and ω. Compare the numerical solution to the exact solution for several choises of parameters. Make sure that at least one of the test involve the boundary conditions in a significant way. By this I mean that $h(t)$ and $k(t)$ should be nonzero for some values of t.

d) Now use you code to solve the problem

$$\begin{aligned} u_{tt} - u_{xx} &= 0 \qquad -l < x < l \\ u(-l,t) &= \sin(\omega t) \\ u(l,t) &= 0 \\ u(x,0) &= 0 \\ u_t(x,0) &= 0 \end{aligned}$$

This model what happend when you periodically move the left end of the string,whose right end is clamped, up and down. What do you expect to happend? Do your code verify your intuition?

12.2 Project2

In this project the aim is to develop a stable numerical scheme for the initial value problem

$$\begin{aligned} au_t + bu_x &= \rho, \qquad -L < x < L, \quad 0 < t < T \\ u(-L,t) &= 0 \\ u(L,t) &= 0 \\ u(x,0) &= \varphi(x) \end{aligned} \tag{799}$$

where

$$\begin{aligned} a &= a(x,t) \\ b &= b(x,t,u) \\ \rho &= \rho(x,t) \end{aligned}$$

Remark: Since the equation is first order in time, one can actually not pose boundary conditions at both endpoints. Thus we should only use the condition at one end point, for example $u(-L,t)=0$. However since we are using center difference we really also need a boundary condition at $x=L$. In this problem I ask you to use the condition $u(-L,t)=0$. Using these two conditions you can construct a finite difference scheme using center difference in space. However, and this is of outmost importance, you must make sure that the solution does not become nonzero in the region close to the boundaries, if it does the scheme will go unstable no matter what we do. But this restriction is really ok since we really are trying to simulate the dynamics on an infinite domain, and thus the computations boundaries at $x=-L,L$ should be "invisible" at all time.

a) Derive a numerical scheme for equation (799) using center differences for the time and space derivatives. Find the Von-Neuman condition for numerical stability of your scheme for the case when $a=a_0, b=b_0$ are constants and $\rho=0$.

b) Verify the numerical stability condition by running the your numerical scheme for some initial condition of your choosing. Run with grid parameters both in the Von Neumann stable and unstable regime Since our numerical scheme is meant to be a good approximation to solutions of equation (799) on an infinite interval, you must make sure that the initial condition is very small or zero close to the end points of the interval $[-L,L]$. A well localized Gaussian function will do the job.

c) Solve the initial value problem

$$\begin{aligned} (3x^2+1)u_t + 2tu_x &= 0, \quad -\infty < x < \infty \\ u(x,0) &= e^{-\gamma x^2} \end{aligned}$$

using the method of characteristics. Plot some of the base characteristic curves in the $x-t$ plane and also plot the solution $u(x,t)$ as a function of x for several $t>0$.

d) Solve the initial value problem from c) using your finite difference scheme. Compare with the solution exact solution from c) by plotting them on top of each other for several times.

e) Let the following quasi linear initial value problem be given

$$\begin{aligned} u_t + uu_x &= 0, \qquad -\infty < x < \infty \\ u(x,0) &= \varphi(x) \end{aligned}$$

Find exact formulas for the shock time for solutions to the initial value problem from e) corresponding to the initial data

i)

$$\varphi(x) = \begin{cases} 0, & x > 1 \\ 1-x, & 0 < x < 1 \\ x+1, & -1 < x < 0 \\ 0, & x < 0 \end{cases}$$

ii)

$$\varphi(x) = \frac{1}{1+x^2}$$

iii)

$$\varphi(x) = sech(x)$$

Test the validity of your finite difference code, and exact formulas for the shock time, by running your code with the given initial conditions and verifying that the graph of the numerical solutions become approximately vertical for some x as the shock time is approached.

12.3 Project3

In this project we are going to solve a heat equation numerically using the finite difference method and the finite Fourier transform.

$$\begin{aligned} u_t - u_{xx} &= \rho(x,t)\ , 0 < x < l\ ,\ t > 0 \\ u(0,t) &= f(t) \\ u(l,t) &= g(t) \\ u(x,0) &= \varphi(x) \end{aligned} \tag{800}$$

a) Implement a finite difference method for this problem. The code should be flexible enough to handle arbitrary choises for the functions f, g, ρ and φ.

b) The function

$$u_e(x,t) = \frac{1}{l}(xh(x,t)(\frac{g(t)}{h(l,t)}) + (l-x)h(x,t)(\frac{f(t)}{h(0,t)})$$

satisfy the boundary conditions. Calculate the corresponding artificial source ρ and initial condition φ. Now run your finite difference code with the calculated ρ and φ and compare the numerical solution to the exact solution $u_e(x,t)$. Do this with a couple of choises for the functions $f(t), g(t)$ and $h(x,t)$ and for several values of the parameters in the problem. For each choise plot the numerical and exact solution in the same plot and thereby verify that they coincide.

c) Solve (800) using the finite Fourier transform. Before you apply the finite Fourier transform you should convert (800) into a problem with homogenous boundary conditions like in equation (434) in the lecture notes. Use the same choices for $f(t)$ and $g(t)$ that you used in problem b) and choose your own source. Compare the finite Fourier transform solution with the finite difference solution for some choises of initial conditions. Make sure that the initial conditions is consistent with the boundary conditions

$$\begin{aligned} \varphi(0) &= f(0) \\ \varphi(l) &= g(0) \end{aligned}$$

d) Let the source in problem (800) be the nonlinear source from equation (445) in the lecture notes.

$$\rho(x,t) = \widehat{\lambda} u(x,t)(1-u^2(x,t))$$

Use the finite difference code to verify the linear and nonlinear stability results found in section 4.7 of our textbook. In particular you should shown that

i) $u(x,t) = 0$ is stable for $\widehat{\lambda} < 1$

ii) $u(x,t)$ is unstable for $1 < \widehat{\lambda} < 4$ and approach the solution 4.7.24 as time increase to infinity.

Run the finite difference code for values of $\widehat{\lambda}$ higher than 4 and plot the resulting solutions. Any conclusions?

References

[1] "Partial Differential Equations of Applied Mathematics", Erich Sauderer, 2006, Wiley.

[2] "A first Course in Partial Differential Equations",H. F. Weinberger, 1965, John Wiley and Sons.

[3] "Introduction to Partial Differential Equations", Gerald B. Folland,1976, Princeton University Press.

[4] "Partial Differential Equations, An Introduction",Walter A. Strauss,2007, John Wiley and Sons.

[5] "Methods of mathematical Physics, Volume II", Courant and Hilbert,1962, Interscience Publishers.

[6] "Partial Differential Equations", Fritz John,1982,Springer.

[7] "Methods of Applied Mathematics", James P. Keener, 1988,Springer.

[8] "Green's functions and Boundary value problems", Ivar Stakgold and Michael J. Holst,2011, Wiley.