Implicit Scheme for Wave Equation

1 Motivation

We want to work on an implicit scheme to solve 1+1 evolution equations as given in [1]. The code for evolving the evolution equations is based on explicit timestepping methods, for which the timestep size is limited by the smallest spatial scale through the Courant-Friedrichs-Lewy (CFL) condition. So in non-linear evolutions this explicit scheme can break down due to violation of CFL condition. Implicit timestepping methods allow for larger timesteps, and can be used to overcome any possible issue due to the CFL condition. We want to develop an implicit scheme for evolution equations in [1] and test our code. In order to reach our goal we started with a simple case to build a foundation. We will first create a code in python and c++ to solve second order 1 dimensional wave equation without a source term.

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \, \frac{\partial^2 u(x,t)}{\partial x^2} \tag{1}$$

with simplest initial consitions

$$u[x_i, t=0] = f(x_i) = \sin(x_i/L) \quad \text{and} \quad \frac{\partial u[x_i, t=0]}{\partial t} = g(x_i) = 0 \tag{2}$$

and boundary conditions

$$u[x=0] = 0 \quad \text{and} \quad u[x=L, t] = 0 \tag{3}$$

2 Review and Algebra for Explicit and Implicit Schemes

I will write finite difference schemes formula for Eq (1) here and will write all steps if needed to verify the expressions below

For explicit centered difference for second derivatives in time and space we have

$$u_i^{n+1} = 2\, u_i^n - u_i^{n-1} + \frac{c^2 (dt)^2}{dx^2}\left(u_{i+1}^n - 2\, u_i^n + u_{i-1}^n\right) \tag{4}$$

For implicit scheme we first use finite difference formula at time $n+1$ rather than n

For space derivative we get

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^{n+1} = \frac{u_{i+1}^{n+1} - 2u_i^{n+1} - u_{i-1}^{n+1}}{(dx)^2}$$

and for time derivative we use two backward time Taylor series expansion

$$u_i^n = u_i^{n+1} - dt\left.\frac{\partial u}{\partial t}\right|_i^{n+1} + \frac{(dt)^2}{2!}\left.\frac{\partial^2 u}{\partial t^2}\right|_i^{n+1} - \frac{(dt)^3}{3!}\left.\frac{\partial^3 u}{\partial t^3}\right|_i^{n+1} + O(dt)^4$$

$$u_i^{n-1} = u_i^{n+1} - 2\, dt\left.\frac{\partial u}{\partial t}\right|_i^{n+1} + \frac{(2dt)^2}{2!}\left.\frac{\partial^2 u}{\partial t^2}\right|_i^{n+1} - \frac{(2dt)^3}{3!}\left.\frac{\partial^3 u}{\partial t^3}\right|_i^{n+1} + O(dt)^4$$

Multiply first equation by 2 and subtract the second equation we have

$$2u_i^n - u_i^{n-1} = u_i^{n+1} - (dt)^2\left.\frac{\partial^2 u}{\partial t^2}\right|_i^{n+1} + O(dt)^4$$

hence we have

$$\left.\frac{\partial^2 u}{\partial t^2}\right|_i^{n+1} = \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{(dt)^2}$$

Plugging into Eq (1)

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{(dt)^2} = c^2 \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(dx)^2}$$

and after rearranging terms we will get

$$\left(1 + 2\frac{c^2(dt)^2}{dx^2}\right) u_i^{n+1} - \frac{c^2(dt)^2}{dx^2} u_{i+1}^{n+1} - \frac{c^2(dt)^2}{dx^2} u_{i-1}^{n+1} = 2\, u_i^n - u_i^{n-1}$$

We will also need the initial condition $\frac{\partial u}{\partial t}\Big|_i^{n=0}$ in implicit form and for that we will follow

In order to get time derivatives in equation we use backward Taylor expansions

$$u_i^n = u_i^{n+1} - dt\, \frac{\partial u}{\partial t}\Big|_i^{n+1} + \frac{(dt)^2}{2!} \frac{\partial^2 u}{\partial t^2}\Big|_i^{n+1} - \frac{(dt)^3}{3!} \frac{\partial^3 u}{\partial t^3}\Big|_i^{n+1} + O(dt)^4 \tag{5}$$

We can also Taylor expand the first time derivative at n around $n+1$

$$\frac{\partial u}{\partial t}\Big|_i^{n} = \frac{\partial u}{\partial t}\Big|_i^{n+1} - dt \frac{\partial^2 u}{\partial t^2}\Big|_i^{n+1} + \frac{(dt)^2}{2!} \frac{\partial^3 u}{\partial t^3}\Big|_i^{n+1} + O(dt)^3$$

which can be written as

$$\frac{\partial u}{\partial t}\Big|_i^{n+1} = \frac{\partial u}{\partial t}\Big|_i^{\mathbf{n}} + dt \frac{\partial^2 u}{\partial t^2}\Big|_i^{n+1} - \frac{(dt)^2}{2!} \frac{\partial^3 u}{\partial t^3}\Big|_i^{n+1} + O(dt)^3$$

and plug into (5) we have

$$u_i^n = u_i^{n+1} - dt \left(\frac{\partial u}{\partial t}\Big|_i^{\mathbf{n}} + dt \frac{\partial^2 u}{\partial t^2}\Big|_i^{n+1} - \frac{(dt)^2}{2!} \frac{\partial^3 u}{\partial t^3}\Big|_i^{n+1} \right) + \frac{(dt)^2}{2!} \frac{\partial^2 u}{\partial t^2}\Big|_i^{n+1} - \frac{(dt)^3}{3!} \frac{\partial^3 u}{\partial t^3}\Big|_i^{n+1}$$

which gives

$$\frac{\partial^2 u}{\partial t^2}\Big|_i^{n+1} = \frac{2}{(dt)^2} \left(u_i^{n+1} - dt \frac{\partial u}{\partial t}\Big|_i^{\mathbf{n}} - u_i^n \right) + O(dt)$$

initially at $n = 0$ we have $u_i^0 = f_i$ and $\frac{\partial u}{\partial t}\Big|_i^{\mathbf{0}} = g_i$. we can use wave equation $\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}$ and hence

$$\frac{\partial^2 u}{\partial x^2}\Big|_i^{n+1} = \frac{u_{i+1}^{n+1} - 2u_i^{n+1} - u_{i-1}^{n+1}}{(dx)^2}$$

for LHS and at $n = 0$ we will finally lines)

$$c^2 \frac{u_{i+1}^1 - 2u_i^1 - u_{i-1}^1}{(dx)^2} = \frac{2}{(dt)^2} \left(u_i^1 - dt \frac{\partial u}{\partial t}\Big|_i^{\mathbf{0}} - u_i^0 \right)$$

$$u_{i+1}^1 - 2u_i^1 - u_{i-1}^1 = \frac{2c^2(dx)^2}{(dt)^2} \left(u_i^1 - dt g_i - f_i \right)$$

which gives

$$\left(1 + \frac{c^2(dt)^2}{dx^2}\right) u_i^1 - \frac{c^2(dt)^2}{2\, dx^2} u_{i+1}^1 - \frac{c^2(dt)^2}{2\, dx^2} u_{i-1}^1 = f_i + dt\, g_i$$

Hence we get this algorithm

- Set up initial conditions at all x points at $t = 0,\ u_i^{n=0} = f(x_i)$
- Calculate u_i at $t = 1$ or $n = 1$ solving implicit set of equations using tridiagnol method and initial conditions

$$\left(1 + \frac{c^2(dt)^2}{dx^2}\right) u_i^{n+1} - \frac{c^2(dt)^2}{2\, dx^2} u_{i+1}^{n+1} - \frac{c^2(dt)^2}{2\, dx^2} u_{i-1}^{n+1} = f_i + dt\, g_i$$

- Proceeding to next time step and compute u_i^{n+1} for subsequent times following implicit set of equations via tridiagonal method and previous time step values while at each subsequent time we use two back time steps

$$\left(1 + 2\,\frac{c^2(dt)^2}{dx^2}\right) u_i^{n+1} - \frac{c^2(dt)^2}{dx^2} u_{i+1}^{n+1} - \frac{c^2(dt)^2}{dx^2} u_{i-1}^{n+1} = 2\,u_i^n - u_i^{n-1}$$

For implicit scheme I will use Tridiagnol solver method. The plots for some times are shown below for my simplest example

videolink

References

[1] Miguel Bezares, Marco Crisostomi, Carlos Palenzuela, and Enrico Barausse. K-dynamics: well-posed 1+1 evolutions in K-essence. *JCAP*, 03:072, 2021.