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

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

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

Second order explicit scheme uses centered difference for second derivatives in time and space and discretize form of (1) is

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

Which gives acceptable numerical results if the convergence condition by Courant–Friedrichs–Lewy satisfied in choosing step sizes $(dt,\, dx)$ in mesh grid.

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

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

hence we have

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

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

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

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

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

which can be written as

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

and plug into (5) we have

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

which gives

$$\frac{\partial^2 u}{\partial t^2}\bigg|_i^{n+1} = \frac{2}{(dt)^2}\left(u_i^{n+1} - dt\frac{\partial u}{\partial t}\bigg|_i^{\mathbf{n}} - u_i^n\right) + O(dt) \tag{6}$$

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

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

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

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

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

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

$$\left(1 + 2\,\frac{c^2(dt)^2}{dx^2}\right) u_i^{n+1} - \frac{c^2(dt)^2}{(dx)^2} u_{i+1}^{n+1} - \frac{c^2(dt)^2}{(dx)^2} u_{i-1}^{n+1} = 2\,u_i^n - u_i^{n-1} \tag{7}$$

For implicit scheme I will use Tri-diagnol solver method. The codes are on github and we can add plots to show convergence. Notice from (6) that this implicit scheme is first order. Further we will be using Tridiagonal solver which it self can make numerical errors so we need to take into account those.

3 Stability

We check the Von Neuman stability analysis on our first order implicit scheme

First we used $u_i^n = \epsilon_i^n = g^n \exp(i\,\theta_n x_i)$ Now using this in our scheme Eq (7) and simplfy a bit we get

$$g_n - 2 + g_n^{-1} = \frac{c^2\,(dt)^2}{(dx)^2} g_n \left[\exp(i\,\theta_n) - 2 + \exp(-i\,\theta_n)\right]$$

which results in quadratic equation

$$\left[1 \;+\; 4\,\frac{c^2\,(dt)^2}{(dx)^2}\,\sin^2\left(\frac{\theta_n}{2}\right)\right] g_n^2 - 2g_n + 1 = 0$$

which gives two complex conjugate roots and their product is

$$|g_n| = \sqrt{\frac{1}{1 \;+\; 4\,\frac{c^2\,(dt)^2}{(dx)^2}\,\sin^2\left(\frac{\theta_n}{2}\right)}}$$

4 Second order Implicit Scheme

We can use expression as on page 199-200 of Mitchel's book

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

We will use central difference for second derivative in time and apply an average of $n+1$ and $n-1$ time on central difference scheme of spatial derivative to have wave equation in form

Rearranging the terms we get

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

which can be written as

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

For first iteration we need a second order expression using initial condition and simplest choice can be central difference formula for first time derivative which gives

$$\left.\frac{\partial u}{\partial t}\right|_i^{n=0} = \frac{u_i^{n=1} - u_i^{n=-1}}{2\,dt}$$

using initial condition as

$$u_i^{-1} = u_i^1 - 2\,dt\left.\frac{\partial u}{\partial t}\right|_i^{n=0}$$

We need to know u^{-1} values from initial conditions u_i^0 and $\left.\frac{\partial u}{\partial t}\right|_i^{n=0}$ but we do not have u_i^1 so we need some trick here to have similar expression as for $n > 0$ case.

We can try simple backward finite difference like this

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

Now this will give first derivative with truncation error of $\mathcal{O}((dt)^2)$ and with using $\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}$,as we did in first order method, and using central difference formula for it

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

using $n = 0$ we have values of u^0 for each except the $i = -1$ and $i = nx + 1$ which can be called ghosts values in x-grid and it we can discuss how to handle those terms but with that we can get

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

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.