Oblig 2 - INF5620

Henrik Andersen Sveinsson

October 2, 2013

1 Introducing the problem

The mathematical problem to be solved is the two-dimensional wave equation:

$$\frac{\partial^2 u}{\partial t^2} + b\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(q(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) + f(x,y,t) \tag{1}$$

2 Deriving the discrete equations

Inserting finite differences for the derivatives:

$$[D_t D_t u + bD_{2t}u = D_x q D_x u + D_y q D_y u + f]^n_{i,j} \tag{2}$$

We write this out to prepare it for implementation:

$$\begin{aligned}
\frac{u^{n+1}_{i,j} - 2u^n_{i,j} + u^{n-1}_{i,j}}{\Delta t^2} + b\frac{u^{n+1}_{i,j} - u^{n-1}_{i,j}}{2\Delta t} \quad &= \qquad (3)\\
\frac{q^n_{i+\frac{1}{2},j}\left(u^n_{i+1,j} - u^n_{i,j}\right) - q^n_{i-\frac{1}{2},j}\left(u^n_{i,j} - u^n_{i-1,j}\right)}{\Delta x^2} & \qquad (4)\\
+\frac{q^n_{i,j+\frac{1}{2}}\left(u^n_{i,j+1} - u^n_{i,j}\right) - q^n_{i,j-\frac{1}{2}}\left(u^n_{i,j} - u^n_{i,j-1}\right)}{\Delta y^2} + f^n_{i,j} & \qquad (5)
\end{aligned}$$

Where the $q_{\frac{1}{2}}$'s will be evaluated with an arithmetic mean. We assume that $\Delta x = \Delta y$ for a simpler scheme.

Now, we write out the arithmetic means for $q_{\frac{1}{2}}$ and isolate $u^n_{i,j}$. (With sympy)

$$\begin{aligned}
u^{n+1}_{i,j} \quad &= \quad \frac{\Delta t^2}{\Delta x^2\left(\Delta t b + 2\right)}\left(q^n_{i+1,j}(u^n_{i+1,j} - u^n_{i,j}) + q^n_{i,j+1}(u^n_{i,j+1} - u^n_{i,j}) + q^n_{i,j-1}(u^n_{i,j-1} - u^n_{i,j})\right.\\
&+ \quad q^n_{i,j}(u^n_{i+1,j} + u^n_{i,j+1} + u^n_{i,j-1} - 4u^n_{i,j} + u^n_{i-1,j})\\
&+ \quad \left. q^n_{i-1,j}(-u^n_{i,j} + u^n_{i-1,j}) + \frac{\Delta x^2}{\Delta t^2}\left[\Delta t b u^{n-1}_{i,j} - 2u^{n-1}_{i,j} + 4u^n_{i,j}\right]\right) + \frac{2\Delta t^2}{(b\Delta t + 2)}f^n_{i,j}
\end{aligned}$$

3 Domain and boundary condition

The partial differential equation is to be solved on a domain $\Omega = [0, L_x] \times [0, L_y]$, which will be discretized by a mesh (i, j) where $i \in (0, N_x)$ and $j \in (0, N_y)$. The boundary condition is $\frac{\partial u}{\partial n} = 0$ where $n \in \{x, y\}$. The numerical scheme as it is formulated now, will required to evaluate points outside the mesh in order to calculate the soultion on the boundary. However, with a symmetric nearest neighbor finite difference, the boundary condition requires $u^n_{i,j} = u_{i+1,j}$ for the boundary at $x = L_x$, similarly for the other boundaries. In the implementation, these points outsie the mesh are implmemented as ghost cells, and they are updated after all other calculations in that timestep.

4 Special formula for the first timestep