INF5620 - Compulsory project 1
Andreas Thune
01.10.2015

**Exercise 2.2**

Want to discretizise the equation (1):

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

on the domain $\Omega = [0, L_x] \times [0, L_y]$, and time in $t \in [0, T]$.

## Discretize domain

Divide $[0, L_x]$ into $N_x$ intervals $[x_i, x_{i+1}]$, with $i \in \{0, ..., N_x - 1\}$, with $x_0 = 0$ and $x_{N_x} = L_x$. Do the same for $[0, L_y]$, so you get $\{y_j\}_{j=0}^{N_y}$, and $[0, T]$ so you get $\{t_n\}_{n=0}^{N_t}$. From now on let $u(x_i, y_j, t_n) = u_{i,j}^n$.

## Discretize equation for inner points

I discretize with finite differences term by term. For $\frac{\partial^2 u}{\partial^2 t}$ I use centred difference:

$$\frac{\partial^2 u}{\partial^2 t}(x_i, y_j, t_n) \approx \frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}}{\Delta t^2}$$

For $b\frac{\partial u}{\partial t}$ I use forward difference:

$$b\frac{\partial u}{\partial t}(x_i, y_j, t_n) \approx b\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}$$

For $\frac{\partial}{\partial x}(q(x,y)\frac{\partial u}{\partial x})$ and $\frac{\partial}{\partial y}(q(x,y)\frac{\partial u}{\partial y})$ I use the normal centred difference for variable coeffichent:

$$\frac{\partial}{\partial x}(q(x,y)\frac{\partial u}{\partial x}(x_i, y_j, t_n)) \approx \frac{1}{\Delta x^2}(q_{i+\frac{1}{2},j}(u_{i+1,j}^n - u_{i,j}^n) - q_{i-\frac{1}{2},j}(u_{i,j}^n - u_{i-1,j}^n))$$

$$\frac{\partial}{\partial y}(q(x,y)\frac{\partial u}{\partial y}(x_i, y_j, t_n)) \approx \frac{1}{\Delta y^2}(q_{i,j+\frac{1}{2}}(u_{i,j+1}^n - u_{i,j}^n) - q_{i,j-\frac{1}{2}}(u_{i,j}^n - u_{i,j-1}^n))$$

Want to find expression for $u_{i,j}^{n+1}$, for calculating inner points. Set $C = \frac{\partial}{\partial x}(q(x,y)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(q(x,y)\frac{\partial u}{\partial y}) + f(x,y,t)$. Then :

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}}{\Delta t^2} + b\frac{u_{i,j}^{n+1} - u_{i,j}^n +}{\Delta t} = C$$

$$\iff u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1} + \Delta t b(u_{i,j}^{n+1} - u_{i,j}^n) = \Delta t^2 C$$

$$\iff u_{i,j}^{n+1}(1+\Delta tb) = u_{i,j}^{n}(2+\Delta tb) - u_{i,j}^{n-1} + \Delta t^2 C$$

$$\iff u_{i,j}^{n+1} = \frac{u_{i,j}^{n}(2+\Delta tb) - u_{i,j}^{n-1} + \Delta t^2 C}{(1+\Delta tb)}$$

If we set in expression for $C$ we get (2):

$$u_{i,j}^{n+1} = \frac{u_{i,j}^{n}(2+\Delta tb) - u_{i,j}^{n-1} + \Delta t^2 f_{i,j}^{n}}{(1+\Delta tb)}$$

$$+\frac{\Delta t^2}{\Delta x^2}\frac{(q_{i+\frac{1}{2},j}(u_{i+1,j}^{n} - u_{i,j}^{n}) - q_{i-\frac{1}{2},j}(u_{i,j}^{n} - u_{i-1,j}^{n}))}{(1+\Delta tb)}$$

$$+\frac{\Delta t^2}{\Delta y^2}\frac{(q_{i,j+\frac{1}{2}}(u_{i,j+1}^{n} - u_{i,j}^{n}) - q_{i,j-\frac{1}{2}}(u_{i,j}^{n} - u_{i,j-1}^{n}))}{(1+\Delta tb)}$$

## Discretize first step

For the first step we do not know $u_{i,j}^{-1}$, but the initial condition $u(x,y.0) = I(x,y)$ gives us $u_{i,j}^{0}$. The second initial condition is $u_t(x,y,0) = V(x,y)$. To find an expression for $u_{i,j}^{-1}$, we discretize $u_t(x,y,0) = V(x,y)$:

$$u_t(x_i,y_j,0) \approx \frac{u_{i,j}^{1} - u_{i,j}^{-1}}{2\Delta t}$$

Set this equal to $V_{i,j}$, and we get an expression for $u_{i,j}^{-1}$:

$$\frac{u_{i,j}^{1} - u_{i,j}^{-1}}{2\Delta t} = V_{i,j} \iff u_{i,j}^{-1} = u_{i,j}^{1} - 2\Delta t V_{i,j}$$

Set this into the above result for inner points we get:

$$u_{i,j}^{1}(1+\Delta tb) = u_{i,j}^{0}(2+\Delta tb) - u_{i,j}^{1} + 2\Delta t V_{i,j} + \Delta t^2 C$$

$$\iff u_{i,j}^{1}(2+\Delta tb) = u_{i,j}^{0}(2+\Delta tb) + 2\Delta t V_{i,j} + \Delta t^2 C$$

$$\iff u_{i,j}^{1} = \frac{u_{i,j}^{0}(2+\Delta tb) + 2\Delta t V_{i,j} + \Delta t^2 C}{2+\Delta tb}$$

Here $C$ is the decrete version of:

$$C = \frac{\partial}{\partial x}(q(x_i,y_j)\frac{\partial u(x_i,y_j,0)}{\partial x}) + \frac{\partial}{\partial y}(q(x_i,y_j)\frac{\partial u(x_i,y_j,0)}{\partial y}) + f(x_i,y_j,0)$$

## Discretize Neumann conditions

Lastly we need to handle the boundary, with homogeneous Neumann conditions. There are three cases:

(i): $(x_i, y_j) \in (0, L_x) \times \{0\} \cup (0, L_x) \times \{L_y\}$.

(ii): $(x_i, y_j) \in \{0\} \times (0, L_y) \cup \{L_x\} \times (0, L_y)$.

(iii): $(x_i, y_j) \in \{(0,0), (0, L_y), (L_x, 0), (L_x, L_y)\}$.

For case (i) I only look at points $(x_i, y_0)$. The Neumann condition gives us $u_y(x_i, y_0, t_n) = 0$. With the normal discretization of this we get:

$$\frac{u^n_{i,1} - u^n_{i,-1}}{2\Delta y} = 0 \iff u^n_{i,-1} = u^n_{i,1}$$

The only part of the scheme that is affected is $\frac{\partial}{\partial y}(q(x,y)\frac{\partial u}{\partial y})$, we get:

$$\frac{1}{\Delta y^2}(q_{i,\frac{1}{2}}(u^n_{i,1} - u^n_{i,0}) - q_{i,-\frac{1}{2}}(u^n_{i,0} - u^n_{i,-1})) = \frac{1}{\Delta y^2}(q_{i,\frac{1}{2}}(u^n_{i,1} - u^n_{i,0}) + q_{i,-\frac{1}{2}}(u^n_{i,1} - u^n_{i,0}))$$

$$= \frac{1}{\Delta y^2}(u^n_{i,1} - u^n_{i,0})(q_{i,\frac{1}{2}} + q_{i,-\frac{1}{2}})$$

Since $q_{i,-\frac{1}{2}}$ is evaluating q outside of our domain, we make the following ansatz: $q_{i,-\frac{1}{2}} \approx q_{i,\frac{1}{2}} \approx q_{i,0}$. This gives:

$$\frac{\partial}{\partial y}(q(x_i, 0)\frac{\partial u(x_i, 0, t_n)}{\partial y}) \approx \frac{2(u^n_{i,1} - u^n_{i,0})q_{i,0}}{\Delta y^2}$$

For $y = L_y$, we get the following by same arguments:

$$\frac{\partial}{\partial y}(q(x_i, L_y)\frac{\partial u(x_i, y_{N_y}, t_n)}{\partial y}) \approx \frac{2(u^n_{i,N_y-1} - u^n_{i,N_y})q_{i,N_y}}{\Delta y^2}$$

For case (ii), we get the same, but now we have:

$$u_x(x_0, y_j, t_n) = u_x(x_{N_x}, y_j, t_n) = 0$$

By the same approach as for case (i), we get

$$\frac{\partial}{\partial x}(q(0, y_j)\frac{\partial u(0, y_j, t_n)}{\partial x}) \approx \frac{2(u^n_{1,j} - u^n_{0,j})q_{0,j}}{\Delta x^2}$$

and

$$\frac{\partial}{\partial x}(q(L_x, y_j)\frac{\partial u(x_{N_x}, y_j, t_n)}{\partial x}) \approx \frac{2(u^n_{N_x-1,j} - u^n_{N_x,j})q_{N_x,j}}{\Delta x^2}$$

For case (iii), we get both case (i) and (ii). This means we already have the expressions needed to explicitly find u for points in (iii). If you want the formulas for the boundary points, you need to substitute the expressions for the x and y derivatives in (2) with the ones we have found above.

**Exercise 3.1**

Want to prove that a constant solution also solves the discrete equation:

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\Delta t^2} + b\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t}$$

$$= \frac{1}{\Delta x^2}(q_{i+\frac{1}{2},j}(u_{i+1,j}^{n} - u_{i,j}^{n}) - q_{i-\frac{1}{2},j}(u_{i,j}^{n} - u_{i-1,j}^{n}))$$

$$+\frac{1}{\Delta y^2}(q_{i,j+\frac{1}{2}}(u_{i,j+1}^{n} - u_{i,j}^{n}) - q_{i,j-\frac{1}{2}}(u_{i,j}^{n} - u_{i,j-1}^{n}))$$

A constant solution $u(x, y, t) = c$ means that $u_{i,j}^{n} = c$. This means that

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\Delta t^2} + b\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{c - 2c + c}{\Delta t^2} + b\frac{c - c}{\Delta t} = 0$$

and:

$$\frac{1}{\Delta x^2}(q_{i+\frac{1}{2},j}(u_{i+1,j}^{n} - u_{i,j}^{n}) - q_{i-\frac{1}{2},j}(u_{i,j}^{n} - u_{i-1,j}^{n}))$$

$$= \frac{1}{\Delta x^2}(q_{i+\frac{1}{2},j}(c - c) - q_{i-\frac{1}{2},j}(c - c)) = 0$$

and:

$$\frac{1}{\Delta y^2}(q_{i,j+\frac{1}{2}}(u_{i,j+1}^{n} - u_{i,j}^{n}) - q_{i,j-\frac{1}{2}}(u_{i,j}^{n} - u_{i,j-1}^{n})) = 0$$

This means that the discrete equation is satisfied when u is constant.

**Exercise 3.1.4: Making bugs**

To make it simple, all my bugs are made in the first step, and are shown in the code.

(1): The first bug is to not implement the initial condition correctly. This causes errors.

(2): The second bug, is to change a sign in the scheme. This does not cause errors, since the terms I subtract are both zero.

(3): The third bug is also a changed sign, but this time the two non-zero terms are added instead of subtracted, and we therefore get errors.

(4): The fourth bug is changing the q function in the in the scheme. This does not create errors, since the constant solution does not depend on q because q is multiplied by zero.

(5): Use current timestep to calculate one of the boarders instead of the previous one. This also causes no error.

**Exercise 3.3: Plug wave solution**
When I tested that i should get exact solution, I made sure that the wave did not hit the boundary at the end time. This means that the exact solution at time t is given by Dalambert formula $u(x,y,t) = \frac{1}{2}(I(x-t,y) + I(x+t,y))$. To check that I got exact solution look in code.

**Exercise 3.4: Standing undamped waves**
Everything is in the code.

**Exercise 3.6: Manufactured solution**
Everything in code. Used sympy to find sorce-term. Did not get exactly 2 in convergence rate, but closer to 2 than 1.