Fys4150
Project 5

Peter Killingstad and Karl Jacobsen

https://github.com/kaaja/fys4150

November 23, 2017

Note to instructurs reagarding Github repository

If the above Github-link does not work, it is eighter because you have not yet accepted our invite to the repository, or you have not yet provided us with an e-mail addres available at Github so that we can invite you. The Github user you will be invited from is "kaaja". If the latter applies to you, please send us an e-mail with an e-mailadress available in Github or your Github username so that we can send you an invite. Our e-mailadresses: peter.killingstad@hotmail.com, karljaco@gmail.com.

1 5a

1.1 Derivation of schemes with truncation errors

All the schemes will be derived from Taylor series expansions, and the truncation error will be related to the remainder in the Taylor-series expansions. This remainded is the error we get when truncating the series by leaving out the remainder.

1.2 Forward Euler

For the time derivative, we expand $u(x, t + \Delta t)$ around t

$$u(x, t + \Delta t) = u(x, t) + u_t(x, t)\Delta t + \mathcal{O}(\Delta t^2) \tag{1a}$$

$$\rightarrow u_t(x, t) = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} + \mathcal{O}(\Delta t) \tag{1b}$$

The space derivative, which is a 2nd derivative, we derive by combining two Taylor series'

$$u(x + \Delta x, t) = u(x, t) + u_x(x, t)\Delta x + \frac{u_{xx}(x, t)\Delta x^2}{2} + \frac{u_{xxx}(x, t)\Delta x^3}{6} + \mathcal{O}(\Delta x^4) \tag{2a}$$

$$u(x - \Delta x, t) = u(x, t) - u_x(x, t)\Delta x + \frac{u_{xx}(x, t)\Delta x^2}{2} - \frac{u_{xxx}(x, t)\Delta x^3}{6} + \mathcal{O}(\Delta x^4) \tag{2b}$$

Now we add (2a) and (2b) and solve for $u_{xx}(x,t)$

$$\begin{aligned}
\Big(u(x+\Delta x,t)+u(x-\Delta x,t)\Big) &= \Big(u(x,t)+u(x,t)\Big) \\
&+ \Big(u_x(x,t)\Delta x + (-u_x(x,t)\Delta x)\Big) \\
&+ \Big(\frac{u_{xx}(x,t)\Delta x^2}{2} + \frac{u_{xx}(x,t)\Delta x^2}{2}\Big) \qquad (3a)\\
&+ \Big(\frac{u_{xxx}(x,t)\Delta x^3}{6} + (-\frac{u_{xxx}(x,t)\Delta x^3}{6})\Big) \\
&+ \Big(\mathcal{O}(\Delta x^4) + \mathcal{O}(\Delta x^4)\Big) \\
&= 2u(x,t) + u_{xx}(x,t)\Delta x^2 + \mathcal{O}(\Delta x^4) \qquad (3b)\\
\rightarrow u_{xx}(x,t) &= \frac{u(x-\Delta x,t) - 2u(x,t) + u(x+\Delta x,t)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \qquad (3c)
\end{aligned}$$

Combining (1b) and (3c) we get the Forward Euler scheme

$$u_t(x,t) = u_{xx}(x,t) \qquad (4a)$$

$$\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} + \mathcal{O}(\Delta t) = \frac{u(x-\Delta x,t) - 2u(x,t) + u(x+\Delta x,t)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \qquad (4b)$$

From (4b) we see that the scheme has a truncation error that goes like $\mathcal{O}(\Delta t)$ in time and $\mathcal{O}(\Delta x^2)$ in space.

We will now analyze the stability of the Forward Euler scheme (4b) by applying Neuman stability analyzis. From the analytical solution of the problem, we know that the particular solutions are on the form $u = e^{-(k\pi)^2 t}e^{ik\pi x}$, where k is an integer greater than one. We observe that the solutions are stable in t, meaning that the solutions do not blow up as t increaes. Based on the analytical particular solution, we make the numerical ansatz

$$u = a_k^n e^{ik\pi x_j} \qquad (5)$$

For the numerical ansatz (5) to reproduce the characteristics of the analytical particular solution, with stability in t, we observe that $|a_k^n| < 1$ in necessary. We now plug in the ansatz (5) into the (4b) and derive an equation for $|a_k^n|$:

$$\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} = \frac{u(x-\Delta x,t) - 2u(x,t) + u(x+\Delta x,t)}{\Delta x^2} \qquad (6a)$$

$$\frac{a_k^{n+1}e^{ik\pi(j+1)\Delta x} - a_k^n e^{ik\pi j\Delta x}}{\Delta t} = \frac{a_k^n e^{ik\pi(j-1)\Delta x} - 2a_k^n e^{ik\pi j\Delta x} + a_k^n e^{ik\pi(j+1)\Delta x}}{\Delta x^2} \qquad (6b)$$

$$a_k^n e^{ik\pi j\Delta x}\ \frac{a_k - 1}{\Delta t} = a_k^n e^{ik\pi j\Delta x}\ \frac{e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x}}{\Delta x^2} \qquad (6c)$$

$$\frac{a_k - 1}{\Delta t} = \frac{e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x}}{\Delta x^2} \qquad (6d)$$

$$a_k = 1 + \frac{\Delta t}{\Delta x^2}(e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x}) \qquad (6e)$$

$$= 1 + \frac{\Delta t}{\Delta x^2}\Big(2\cos(k\pi\Delta x) - 2\Big) \qquad (6f)$$

$$= 1 + 2\frac{\Delta t}{\Delta x^2}\Big(\cos(k\pi\Delta x) - 1\Big) \qquad (6g)$$

$$= 1 + 2\frac{\Delta t}{\Delta x^2}\Big(-2\sin^2(\frac{k\pi\Delta x}{2})\Big) \qquad (6h)$$

$$= 1 - 4\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2}) \qquad (6i)$$

$$|a_k| = |1 - 4\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})| \qquad (6j)$$

From (6j) we get

$$|a_k| < 1 \text{ if } ||1 - 4\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})|| < 1 \tag{7a}$$

$$\rightarrow |1 - 4\frac{\Delta t}{\Delta x^2}| < 1 \rightarrow |a_k| < 1 \text{ (Since } \sin^2(k\pi\Delta x/2)_{max} = 1) \tag{7b}$$

$$\rightarrow 1 - 4\frac{\Delta t}{\Delta x^2} > -1 \tag{7c}$$

$$\rightarrow \frac{\Delta t}{\Delta x^2} < \frac{1}{2} \tag{7d}$$

(7d) gives that the Forward Euler scheme is conditionally stable, and the condition that ensures stability.

1.3 Backward Euler

Here we will do the same as we did for Forward Euler above: Derive the scheme, including truncation errors, and analyze stability.

The only change compared to Forward Euler, is the time discretization, which now becomes

$$u(x, t - \Delta t) = u(x, t) + u_t(x, t)\Delta t - \mathcal{O}(\Delta t^2) \tag{8a}$$

$$\rightarrow u_t(x, t) = \frac{u(x, t) - u(x, t - \Delta t)}{\Delta t} + \mathcal{O}(\Delta t) \tag{8b}$$

The space discretization is the same as for Forward Euler, (3c). Combining the space discretization (3c) and (8b) gives

$$\frac{u(x, t) - u(x, t - \Delta t)}{\Delta t} + \mathcal{O}(\Delta t) = \frac{u(x - \Delta x, t) - 2u(x, t) + u(x + \Delta x, t)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{9a}$$

We note that the truncation errors have the same asymptoptic behavior as for the Forward Euler scheme.

Now lets check the stability of the Backward Euler scheme. We apply the same method as we did for Forward Euler, and insert the ansatz (5) into (9a) to get

$$\frac{u(x, t) - u(x, t - \Delta t)}{\Delta t} = \frac{u(x - \Delta x, t) - 2u(x, t) + u(x + \Delta x, t)}{\Delta x^2} \tag{10a}$$

$$\frac{a_k^n e^{ik\pi j\Delta x} - a_k^{n-1} e^{ik\pi j\Delta x}}{\Delta t} = \frac{a_k^n e^{ik\pi(j-1)\Delta x} - 2a_k^n e^{ik\pi j\Delta x} + a_k^n e^{ik\pi(1+j)\Delta x}}{\Delta x^2} \tag{10b}$$

$$a_k^n e^{ik\pi j\Delta x}\ \frac{1 - a_k^{-1}}{\Delta t} = a_k^n e^{ik\pi j\Delta x}\ \frac{e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x}}{\Delta x^2} \tag{10c}$$

$$\frac{1 - a_k^{-1}}{\Delta t} = \frac{e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x}}{\Delta x^2} \tag{10d}$$

$$a_k^{-1} = 1 - \frac{\Delta t}{\Delta x^2}(e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x}) \tag{10e}$$

$$a_k = \frac{1}{1 - \frac{\Delta t}{\Delta x^2}(e^{-ik\pi\Delta x} - 2 + e^{ik\pi\Delta x})} \tag{10f}$$

$$\overset{(6)}{=} \frac{1}{1 + 4\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})} \tag{10g}$$

$$|a_k| = \left|\frac{1}{1 + 4\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})}\right| < 1. \tag{10h}$$

$$\tag{10i}$$

From (10h) we see that, in contrast to the Forward Euler scheme, the Backward Euler scheme is unconditionally stable.

(9a) reveals another difference between the Backward Euler scheme and the Forward Euler scheme: (9a) is implicit in $u(x,t)$, meaning that we cannot solve (9a) directly for $u(x,t)$, as we did in the Forward Euler scheme. However, we can find $u(x,t)$ from (9a) by recognizing that (9a) can be rewritten as a linear system:

$$\frac{u(x,t)-u(x,t-\Delta t)}{\Delta t} = \frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2} \tag{11a}$$

$$\frac{u_i^n-u_i^{n-1}}{\Delta t} = \frac{u_{i-1}^n-2u_i^n+u_{i+1}^n}{\Delta x^2} \tag{11b}$$

$$\frac{\Delta x^2}{\Delta t}(u_i^n-u_i^{n-1}) = u_{i-1}^n-2u_i^n+u_{i+1}^n \tag{11c}$$

$$-\Big(u_{i-1}^n-(2+\frac{\Delta x^2}{\Delta t})u_i^n+u_{i+1}^n\Big) = \frac{\Delta x^2}{\Delta t}u_i^{n-1} \tag{11d}$$

$$\Big(-u_{i-1}^n+(2+\frac{\Delta x^2}{\Delta t})u_i^n-u_{i+1}^n\Big) = \frac{\Delta x^2}{\Delta t}u_i^{n-1} \tag{11e}$$

$$\underbrace{\begin{bmatrix} 2+\frac{\Delta x^2}{\Delta t} & -1 & \cdots & 0 \\ -1 & 2+\frac{\Delta x^2}{\Delta t} & -1 & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & -1 & 2+\frac{\Delta x^2}{\Delta t} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_N^n \end{bmatrix}}_{\mathbf{U}} = \underbrace{\frac{\Delta x^2}{\Delta t}\begin{bmatrix} u_0^{n-1} \\ u_1^{n-1} \\ \vdots \\ u_N^{n-1} \end{bmatrix}}_{\tilde{\mathbf{b}}} \tag{11f}$$

We see from (11f) that solving Backward Euler corresponds to solving a linear system $AU=\tilde{b}$, where A is a trdiagonal matrix.

1.4 Crank-Nicolson

Here we Taylor expand $u(x+\Delta x,t+\Delta t)$ and $u(x-\Delta x,t+\Delta t)$ around $t'=t+\Delta t/2$ to get

$$\begin{aligned} u(x+\Delta x,t+\Delta t) = u(x,t') &+ \frac{\partial u(x,t')}{\partial x}\Delta x + \frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{4} \\ &+ \frac{\partial^2 u(x,t')}{\partial x\partial t}\frac{\Delta t}{2}\Delta x + \mathcal{O}(\Delta t^3) \end{aligned} \tag{12a}$$

$$\begin{aligned} u(x-\Delta x,t+\Delta t) = u(x,t') &- \frac{\partial u(x,t')}{\partial x}\Delta x + \frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{4} \\ &- \frac{\partial^2 u(x,t')}{\partial x\partial t}\frac{\Delta t}{2}\Delta x + \mathcal{O}(\Delta t^3) \end{aligned} \tag{12b}$$

$$\begin{aligned} u(x+\Delta x,t) = u(x,t') &+ \frac{\partial u(x,t')}{\partial x}\Delta x - \frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{4} \\ &- \frac{\partial^2 u(x,t')}{\partial x\partial t}\frac{\Delta t}{2}\Delta x + \mathcal{O}(\Delta t^3) \end{aligned} \tag{12c}$$

$$\begin{aligned} u(x-\Delta x,t) = u(x,t') &- \frac{\partial u(x,t')}{\partial x}\Delta x - \frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{4} \\ &+ \frac{\partial^2 u(x,t')}{\partial x\partial t}\frac{\Delta t}{2}\Delta x + \mathcal{O}(\Delta t^3) \end{aligned} \tag{12d}$$

$$u(x,t+\Delta t) = u(x,t') + \frac{\partial u(x,t')}{\partial t}\frac{\Delta_t}{2} + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2 + \mathcal{O}(\Delta t^3) \tag{12e}$$

$$u(x,t) = u(x,t') - \frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2 + \mathcal{O}(\Delta t^3) \tag{12f}$$

The above formulae are taken from Hjorth-Jensen's slides [2].

Combining (12e) and (12f) gives the time derivative

$$\begin{aligned}\Big(u(x,t+\Delta t)-u(x,t)\Big) &= \Big(u(x,t')-u(x,t')\Big)+\Big(\frac{\partial u(x,t')}{\partial t}\frac{\Delta_t}{2}-(-\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2})\Big)\\ &\quad+\Big(\frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2-\frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2\Big)+\Big(\mathcal{O}(\Delta t^3)-\mathcal{O}(\Delta t^3)\Big)\end{aligned} \tag{13a}$$

$$u(x,t+\Delta t)-u(x,t) = \frac{\partial u(x,t')}{\partial t}\Delta t+\mathcal{O}(\Delta t^3) \tag{13b}$$

$$\frac{\partial u(x,t')}{\partial t} = \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}+\mathcal{O}(\Delta t^2) \tag{13c}$$

Now for the spacial derivative. First we solve (12a) and (12b) for $u_{xx}(x,t')$

$$\begin{aligned}u(x+\Delta x,t+\Delta t)+u(x-\Delta x,t+\Delta t) &= \Big(u(x,t')+u(x,t')\Big)+\Big(\frac{\partial u(x,t')}{\partial x}\Delta x-\frac{\partial u(x,t')}{\partial x}\Delta x\Big)\\ &\quad+\Big(\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2}+\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{2}\Big)+\Big(\frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2+\frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2\Big)\\ &\quad+\Big(\frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{4}+\frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{4}\Big)+\Big(\frac{\partial^2 u(x,t')}{\partial x\partial t}\frac{\Delta t}{2}\Delta x-\frac{\partial^2 u(x,t')}{\partial x\partial t}\frac{\Delta t}{2}\Delta x\Big)\\ &\quad+\Big(\mathcal{O}(\Delta t^3)+\mathcal{O}(\Delta t^3)\Big)+\mathcal{O}(\Delta x^4)\end{aligned} \tag{14a}$$

$$= 2u(x,t')+\frac{\partial u(x,t')}{\partial t}\Delta t+\frac{\partial^2 u(x,t')}{\partial x^2}\Delta x^2+\frac{\partial^2 u(x,t')}{2\partial t^2}\frac{\Delta t^2}{2}+\mathcal{O}(\Delta t^3)+\mathcal{O}(\Delta x^4) \tag{14b}$$

$$= 2u(x,t')+\frac{\partial u(x,t')}{\partial t}\Delta t+\frac{\partial^2 u(x,t')}{\partial x^2}\Delta x^2+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^4) \tag{14c}$$

$$\begin{aligned}\frac{\partial^2 u(x,t')}{\partial x^2}\Delta x^2 &= u(x+\Delta x,t+\Delta t)+u(x-\Delta x,t+\Delta t)-2u(x,t')\\ &\quad+\frac{\partial u(x,t')}{\partial t}\Delta t+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^4)\end{aligned} \tag{14d}$$

$$\begin{aligned}\frac{\partial^2 u(x,t')}{\partial x^2} &= \frac{u(x-\Delta x,t+\Delta t)-2u(x,t')+u(x+\Delta x,t+\Delta t)}{\Delta x^2}\\ &\quad+\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{\Delta x^2}+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)\end{aligned} \tag{14e}$$

Note that in (14e) we added $\mathcal{O}(\Delta x^4)$, since $\mathcal{O}(\Delta x^3)$ cancells out in the addition. Also note that the Δx^3-term was not originally included in (12a) and (12b), but these would have been of equal magnitude with opposite sign, so when adding the equaions we would have been left with $\mathcal{O}(\Delta x^4)$.

Doing the same as in (14e) for (12c) and (12d) we obtain

$$\begin{aligned}\frac{\partial^2 u(x,t')}{\partial x^2} &= \frac{u(x-\Delta x,t)-2u(x,t')+u(x+\Delta x,t)}{\Delta x^2}\\ &\quad-\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{\Delta x^2}+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)\end{aligned} \tag{15a}$$

Now we take the mean of (14e) and (15)

$$
\begin{aligned}
u_{xx}(x,t') &= \frac{1}{2}\Big(\frac{u(x-\Delta x,t+\Delta t)-2u(x,t')+u(x+\Delta x,t+\Delta t)}{\Delta x^2}+\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{\Delta x^2}+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)\\
&+\frac{u(x-\Delta x,t)-2u(x,t')+u(x+\Delta x,t)}{\Delta x^2}-\frac{\partial u(x,t')}{\partial t}\frac{\Delta t}{\Delta x^2}+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)\Big)\\
&=\frac{1}{2}\Big(\frac{u(x-\Delta x,t+\Delta t)-2u(x,t')+u(x+\Delta x,t+\Delta t)}{\Delta x^2}\\
&+\frac{u(x-\Delta x,t)-2u(x,t')+u(x+\Delta x,t)}{\Delta x^2}\Big)+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)\\
&\underset{=}{u(x,t')=\frac{u(x,t)+u(x,t+\Delta t)}{2}}\\
&\frac{1}{2}\Big(\frac{u(x-\Delta x,t+\Delta t)-u(x,t)+u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2}\\
&+\frac{u(x-\Delta x,t)-u(x,t)+u(x,t+\Delta t)+u(x+\Delta x,t)}{\Delta x^2}\Big)+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)\\
&=\frac{1}{2}\Big(\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2}+\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2}\Big)\\
&+\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^2)
\end{aligned}
\tag{16a}
$$

Now combining (13c) and (16a) we get the Crank-Nicolson scheme

$$
\begin{aligned}
\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}+\mathcal{O}(\Delta t^2) &= \frac{1}{2}\Big(\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2}\\
&+\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2}\Big)+\mathcal{O}(\Delta x^2),
\end{aligned}
\tag{17a}
$$

where we have put both $\mathcal{O}(\Delta t^2)$ into a common term.

(17) shows that the Crank-Nicolson scheme is 2nd order in both time and space, implying better convergence properties for the Crank-Nicolson scheme compared to the Backward Euler scheme and the Forward Euler scheme.

Using the same method as we used for the previous schemes, we now study the stability of the Crank-Nicolson scheme. We insert the ansatz (5) into the Crank-Nicolson scheme (17) and solve for $|a_k|$:

$$
\begin{aligned}
\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} &= \frac{1}{2}\Big(\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2}\\
&+\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2}\Big)\\
a_k^n e^{ik\pi j\Delta x}\ \frac{a_k-1}{\Delta t} &= \frac{a_k^n e^{ik\pi j\Delta x}}{2}\Big(\frac{a_k e^{-ik\pi\Delta x}-2a_k+a_k e^{ik\pi\Delta x}}{\Delta x^2}+\frac{e^{-ik\pi\Delta x}-2+e^{ik\pi\Delta x}}{\Delta x^2}\Big)\\
\frac{a_k-1}{\Delta t} &= \frac{1}{2}\Big(\frac{a_k e^{-ik\pi\Delta x}-2a_k+a_k e^{ik\pi\Delta x}}{\Delta x^2}+\frac{e^{-ik\pi\Delta x}-2+e^{ik\pi\Delta x}}{\Delta x^2}\Big)\\
&= \frac{1+a_k}{2\Delta x^2}(e^{-ik\pi\Delta x}-2+e^{ik\pi\Delta x})\\
&\overset{(6)}{=} -4\frac{1+a_k}{2\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})\\
a_k-1 &= (1+a_k)(-\frac{2\Delta t}{\Delta x^2})\sin^2(\frac{k\pi\Delta x}{2})\\
\Big(1+\frac{2\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})\Big)a_k &= 1-\frac{2\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})\\
a_k &= \frac{1-\frac{2\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})}{1+\frac{2\Delta t}{\Delta x^2}\sin^2(\frac{k\pi\Delta x}{2})}<1
\end{aligned}
\tag{18a, 18b}
$$

(18b) shows that the Crank-Nicolson scheme is unconditionally stable.

Based on the analyzis of the different schemes, we expect the Crank-Nicolson scheme to be the best scheme with respect to convergence and stability.

1.5 θ-rule

All the schemes derived above can be derived from a more general scheme, called the θ-rule scheme. To see this, first notice that the Crank-Nicolson scheme (17) is the average of the Forward Euler scheme (4b) and the Backward Euler scheme (9a). We can think of the average, represented by the Crank-Nicolson scheme, as the special case of equals weights in a weighted average of the Backward Euler scheme and the Forward Euler scheme. Writing the weighted average of the Forward Euler and Backward Euler schemes with weights $0 \leq \theta \leq 1$, we get the θ-rule

$$(1-\theta)\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}+\theta\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}=\theta\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2} + (1-\theta)\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2} \tag{19a}$$

$$\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}=\theta\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2} + (1-\theta)\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2} \tag{19b}$$

From the θ-scheme (19b) we see that $\theta = 0,\ 1/2,\ 1$ corresponds to the Forward Euler, the Crank-Nicolson and the Backward Euler scheme respectively.

When implementing the 1D-schemes, we will use the θ-scheme. Using the θ-scheme, we need only write one scheme instead of three.

1.6 Implementation of 1D problem

As mentioned in the previous paragraph, we implement the 1D schemes using the θ-scheme. We will now rewrite (19b) to a linear system, and then we will apply the Thomas algorithm to this system.

To ease the notation, we introduce

$$\alpha = \frac{\Delta t}{\Delta x^2}. \tag{20}$$

Insertion of α (20) into the θ-scheme (19b) and introdusing the discretization $u(x+\Delta x, t-\Delta t) = u_{i+1}^{n-1}$ gives

$$\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} = \theta\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{\Delta x^2} + (1-\theta)\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{\Delta x^2} \tag{21a}$$

$$u(x,t+\Delta t)-u(x,t) = \alpha\theta\Big(u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)\Big) + \alpha(1-\theta)\Big(u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)\Big) \tag{21b}$$

$$u_i^{n+1}-u_i^n = \alpha\theta\Big(u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}\Big)+\alpha(1-\theta)\Big(u_{i-1}^n-2u_i^n+u_{i+1}^n\Big) \tag{21c}$$

$$u_i^n-u_i^{n-1} = \alpha\theta\Big(u_{i-1}^n-2u_i^n+u_{i+1}^n\Big)+\alpha(1-\theta)\Big(u_{i-1}^{n-1}-2u_i^{n-1}+u_{i+1}^{n-1}\Big) \tag{21d}$$

$$u_i^n-\alpha\theta\Big(u_{i-1}^n-2u_i^n+u_{i+1}^n\Big) = u_i^{n-1}+\alpha(1-\theta)\Big(u_{i-1}^{n-1}-2u_i^{n-1}+u_{i+1}^{n-1}\Big) \tag{21e}$$

$$-\alpha\theta u_{i-1}^n+(2\alpha\theta+1)u_i^n-\alpha\theta u_{i+1}^n = \alpha(1-\theta)u_{i-1}^{n-1}+\Big(1-2\alpha(1-\theta)\Big)u_i^{n-1}+\alpha(1-\theta)u_{i+1}^{n-1} \tag{21f}$$

$$-2\alpha\theta u_{i-1}^n+2(2\alpha\theta+1)u_i^n-2\alpha\theta u_{i+1}^n = 2\alpha(1-\theta)u_{i-1}^{n-1}+2\Big(1-2\alpha(1-\theta)\Big)u_i^{n-1}+2\alpha(1-\theta)u_{i+1}^{n-1} \tag{21g}$$

Now we rewrite (21g) as a matrix-vector equation $A_1U^n = A_2U^{n-1}$:

$$\underbrace{\begin{bmatrix} 2(2\alpha\theta+1) & -2\alpha\theta & \cdots & 0 \\ -2\alpha\theta & 2(2\alpha\theta+1) & -2\alpha\theta & \vdots \\ \vdots & & \ddots & -2\alpha\theta \\ 0 & \cdots & -2\alpha\theta & 2(2\alpha\theta+1) \end{bmatrix}}_{\mathbf{A_1}} \underbrace{\begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_N^n \end{bmatrix}}_{\mathbf{U^n}}$$

$$= \underbrace{\begin{bmatrix} 2(1-2\alpha(1-\theta)) & 2\alpha(1-\theta) & \cdots & 0 \\ 2\alpha(1-\theta) & 2(1-2\alpha(1-\theta)) & 2\alpha(1-\theta) & \vdots \\ \vdots & & \ddots & 2\alpha(1-\theta) \\ 0 & \cdots & 2\alpha(1-\theta) & 2(1-2\alpha(1-\theta)) \end{bmatrix}}_{\mathbf{A_2}} \underbrace{\begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ \vdots \\ u_N^{n-1} \end{bmatrix}}_{\mathbf{U^{n-1}}} \tag{22a}$$

In (22a), the right hand side $\mathbf{A_2U^{n-1}}$ is known, since $\mathbf{U}^{n-1}$ is the previous periods solution, so (22a) is a linear system of the known type $AU = b$. So in principle the above system is solved for each time step, solving it for a time step and then using this solution in the right hand side in the next time step.

We note that with $\theta = 0$ in (22a), the system reduces to the explicit Forward Euler system, which can be obtained from (4b). $\theta = 1$ gives the Backward Euler system (11f). $\theta = 1/2$ in (22a) gives the system that can be obtained by rewrting the Crank-Nicolson scheme (17) as a linear system.

In principle the above system can be solved by calculating the inverse of A_1 and solving $U^n = A_1^{-1}(A_2U^{n-1})$. This is a costly operation and is avoided. Instead of calculating the inverse, one often solves these systems by some kind of elimination method, e.g. Gaussian elimination or by LU. In this case, we note that we are dealing with a tridiagonal matrix. For tridiagonal systems, the Thomas algorithm gives an alterative way of solving the linear system which reduces the number of FLOPS considerably. In our case we are eaven more lucky. Our matrix is symmetric, and the elements are constant, so the standard Thomas algorithm can be further improved. We will now derive the algorithm that we implement.

We start with a tridiagonal symmetric linear system $Au = f$, with A being 4×4. The following show the forward

substitution steps for this sytem

$$\begin{bmatrix} d_1 & e_1 & 0 & 0 & f_1 \\ e_2 & d_2 & e_2 & 0 & f_2 \\ 0 & e_3 & d_3 & e_3 & f_3 \\ 0 & 0 & e_4 & d_4 & f_4 \end{bmatrix} \rightarrow \begin{bmatrix} d_1 & e_1 & 0 & 0 & f_1 \\ 0 & \tilde{d}_2 & e_2 & 0 & \tilde{f}_2 \\ 0 & e_3 & d_3 & e_3 & f_3 \\ 0 & 0 & e_4 & d_4 & f_4 \end{bmatrix} \rightarrow \begin{bmatrix} d_1 & e_1 & 0 & 0 & f_1 \\ 0 & \tilde{d}_2 & e_2 & 0 & \tilde{f}_2 \\ 0 & 0 & \tilde{d}_3 & e_3 & \tilde{f}_3 \\ 0 & 0 & e_4 & d_4 & f_4 \end{bmatrix} \rightarrow \begin{bmatrix} d_1 & e_1 & 0 & 0 & f_1 \\ 0 & \tilde{d}_2 & e_2 & 0 & \tilde{f}_2 \\ 0 & 0 & \tilde{d}_3 & e_3 & \tilde{f}_3 \\ 0 & 0 & e_4 & \tilde{d}_4 & \tilde{f}_4 \end{bmatrix} \tag{23a}$$

From (22a) we have that in our case $e_1 = e_2 = ... = e_N = -2\alpha$ and $d_1 = d_2 = ... = d_N = 2(2\alpha\theta + 1)$. Calculating the first couple of $\tilde{d}$'s we quickly see that we get the following general expression for $\tilde{d}_i$

$$\tilde{d}_i = 2(2\alpha\theta + 1) + \frac{(2\alpha)^2}{\tilde{d}_{i-1}} \text{ for } i > 1. \tag{24}$$

Doing the same for $\tilde{f}$ we get the general expression

$$\tilde{f}_i = f_i + \frac{2\alpha\theta}{\tilde{d}_{i-1}} \tilde{f}_{i-1} \text{ for } i > 1. \tag{25}$$

Having $\tilde{d}$ and $\tilde{f}$ we are ready to do the back substitution step. We have

$$\begin{bmatrix} d_1 & e_1 & 0 & 0 \\ 0 & \tilde{d}_2 & e_2 & 0 \\ 0 & 0 & \tilde{d}_3 & e_3 \\ 0 & 0 & e_4 & \tilde{d}_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \\ \tilde{f}_4 \end{bmatrix} \tag{26a}$$

From (26) we get

$$u_4 = \frac{\tilde{f}_4}{\tilde{d}_4} \tag{27}$$

and

$$u_i = \frac{\tilde{f}_i + 2\alpha\theta u_{i+1}}{\tilde{d}_i} \text{ for } i > 1. \tag{28}$$

The equations (24), (25), (27) and (28) is what we need for our algorithm.

```
for time in times:
   Calculate RHS f (= A_2 U^{n-1})

   // Forward substitution
   for row in rows (Start from 1st row):
      Calculate \tilde{d}
      Calculate \tilde{f}
   end row loop

   // Backward substitution
   For row in rows (Starting from last row)
      Calculate u
   end row loop

   // Update RHS for next time step
   U^{n-1} = U^{n}

end time loop
```

1.7 2D schemes

We will derive and apply a Forward Euler (explicit) scheme and a Backward Euler (implicit) scheme for the 2D case.

1.7.1 2D explicit scheme

The new thing in the 2D case for the explicit scheme, is that we get an extra term for u_{yy} in (4b). u_{yy} is approximated in exactly the same way as u_{xx} (3c), so we get

$$u_{yy}(x,y,t) = \frac{u(x,y-\Delta y,t) - 2u(x,y,t) + u(x,y+\Delta y,t)}{\Delta y^2} + \mathcal{O}(\Delta y^2) \tag{29}$$

Adding (29) to the 1D FE scheme (4b) gives the explicit 2D-scheme

$$\begin{aligned}\frac{u(x,y,t+\Delta t) - u(x,y,t)}{\Delta t} + \mathcal{O}(\Delta t) &= \frac{u(x-\Delta x,y,t) - 2u(x,y,t) + u(x+\Delta x,y,t)}{\Delta x^2} \\ &+ \frac{u(x,y-\Delta y,t) - 2u(x,y,t) + u(x,y+\Delta y,t)}{\Delta y^2} \qquad (30a) \\ &+ \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2) \\ \rightarrow u_{ij}^{n+1} &\overset{(20)}{\approx} u_{ij}^n + \alpha(u_{i-1,j}^n + u_{i+1,j}^n + u_{i,j-1}^n + u_{i,j+1}^n - 4u_{ij}^n), \qquad (30b)\end{aligned}$$

where we in the last transition assumed $\Delta x = \Delta y$ such that α is given as before.

First we observe that the truncation errors go like $\mathcal{O}(\Delta t)$, $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta y^2)$.

Now lets analyze the stability, by inserting the ansatz

$$u = a_k^n e^{ik\pi\Delta xj} e^{ik\pi\Delta yl} \tag{31}$$

into (30b) and solve for $|a|$, as before

$$\begin{aligned}u_{ij}^{n+1} &= u_{ij}^n + \alpha(u_{i-1,j}^n + u_{i+1,j}^n + u_{i,j-1}^n + u_{i,j+1}^n - 4u_{ij}^n) \qquad (32a) \\ a_k^n e^{i\pi k\Delta xj} e^{i\pi k\Delta jl} a &= a_k^n e^{i\pi k\Delta xj} e^{i\pi k\Delta jl}\Big(1 + \alpha(e^{-ik\pi\Delta x} + e^{ik\pi\Delta x}) - 2 + e^{-ik\pi\Delta y} + e^{ik\pi\Delta y}) - 2\Big) \qquad (32b) \\ a &= \Big(1 + \alpha(e^{-ik\pi\Delta x} + e^{ik\pi\Delta x}) - 2 + e^{-ik\pi\Delta y} + e^{ik\pi\Delta y}) - 2\Big) \qquad (32c) \\ &\overset{(6)}{=} 1 - 4\alpha\Big(\sin^2(k\pi\Delta x/2) + \sin^2(k\pi\Delta y/2)\Big) \qquad (32d) \\ |a| &= \Big|1 - 4\alpha\Big(\sin^2(k\pi\Delta x/2) + \sin^2(k\pi\Delta y/2)\Big)\Big| \qquad (32e) \\ |a| < 1 \rightarrow &\Big|1 - 4\alpha\Big(\sin^2(k\pi\Delta x/2) + \sin^2(k\pi\Delta y/2)\Big)\Big| < 1 \qquad (32f) \\ &\Big|1 - 4\alpha\Big(\sin^2(k\pi\Delta x/2) + \sin^2(k\pi\Delta y/2)\Big)\Big| < |1 - 8\alpha| \qquad (32g)\end{aligned}$$

Again assuming $\Delta x = \Delta y$, we have $\alpha = \Delta t/\Delta x^2 > 0$, so that (32f) and (32g) gives

$$\begin{aligned}1 - 8\alpha &< -1 \qquad (33a) \\ \alpha &< \frac{1}{4}. \qquad (33b)\end{aligned}$$

(33b) gives that the 2D Forward Euler scheme is conditionally stable. We note that we could probably have applied a simpler ansatz than (31), since it in (30b) was already assumed that $\Delta x = \Delta y$.

The new thing in (30b) compared to (4b), is that uij^{n+1} in the 2D case is a matrix and not a vector as in the 1D case. The fact that we are dealing with a matrix instead of a vector, does not change anything with regards to the solution strategy. The equation (30b) is still explicit, the only difference in implementation is that instead of one single loop a double loop over two space dimensions is needed now.

The algorithm for solving 2D Forward Euler follows.

```
Set IC

for time in times:
   Set BC

   for x in X:
      for y in Y:
         Calculate u_{ij}^{n+1} = f(u^{n})
      end y-loop
   end x-loop

   Set u^{n} = u^{n+1}

end time-loop
```

1.7.2 2D implicit scheme

We start in the same way as we did for Forward Euler in 2D, and just add an extra term for u_{yy} to the 1D Backward Euler scheme (9a). In addition we will assume $h = \Delta x = \Delta y$.

$$\begin{aligned}\frac{u(x,y,t) - u(x,y,t-\Delta t)}{\Delta t} + \mathcal{O}(\Delta t) &= \frac{u(x-\Delta x,y,t) - 2u(x,y,t) + u(x+\Delta x,y,t)}{\Delta x^2} \\ &+ \frac{u(x,y-\Delta y,t) - 2u(x,y,t) + u(x,y+\Delta y,t)}{\Delta y^2} \\ &+ \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2)\end{aligned} \tag{34a}$$

$$\begin{aligned}\frac{u_{i,j}^n - u_{i,j}^{n-1}}{\Delta t} &\approx \frac{u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n}{\Delta x^2} \\ &+ \frac{u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n}{\Delta y^2}\end{aligned} \tag{34b}$$

$$\begin{aligned}\frac{u_{i,j}^n - u_{i,j}^{n-1}}{\Delta t} &\overset{h=\Delta x=\Delta y}{\approx} \frac{u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n}{h^2} \\ &+ \frac{u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n}{h^2}\end{aligned} \tag{34c}$$

$$u_{i,j}^n - u_{i,j}^{n-1} \overset{(20)}{\approx} \alpha(u_{i-1,j}^n - 4u_{i,j}^n + u_{i+1,j}^n + u_{i,j-1}^n + u_{i,j+1}^n) \tag{34d}$$

$$u_{i,j}^n = u_{i,j}^{n-1} + \frac{1}{1+4\alpha}(u_{i-1,j}^n + u_{i+1,j}^n + u_{i,j-1}^n + u_{i,j+1}^n) \tag{34e}$$

First we observe that the truncation errors go like $\mathcal{O}(\Delta t)$, $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta y^2)$, so there is no difference in the truncation errors between Forward Euler and Backward Euler, just as for the 1D case.

We study stability of the 2D Backward Euler scheme by inerting the 2D ansatz (31) into (34d)

$$u_{i,j}^n - u_{i,j}^{n-1} = \alpha(u_{i-1,j}^n - 4u_{i,j}^n + u_{i+1,j}^n + u_{i,j-1}^n + u_{i,j+1}^n) \tag{35a}$$

$$a_k - 1 = \alpha a(e^{-ik\pi\Delta x} + e^{ik\pi\Delta x} - 2 + e^{-ik\pi\Delta y} + e^{ik\pi\Delta y} - 2) \tag{35b}$$

$$\overset{(6)}{=} -4\alpha a\Big(\sin^2(k\pi\Delta x/2) + \sin^2(k\pi\Delta y/2)\Big) \tag{35c}$$

$$\overset{\Delta x=\Delta y=h}{=} -8\alpha a\sin^2(kh/2) \tag{35d}$$

$$a = \frac{1}{1 + 8\alpha\sin^2(kh/2)} < 1 \tag{35e}$$

From (35e) we see that the Backward Euler scheme is also unconditionally stable in two dimensions, as it was for one dimension. We note that we probably could have applied a simpler ansatz than (31), since it in (34d) was already assumed that $\Delta x = \Delta y$.

We see that (34e) is an implicit scheme, since the only known is $u_{i,j}^n$. (34e) can be transformed into a linear system of the known type $Au = b$. The procedure for transforming (34e) into a linear system is the same as demonstrated for the 1D case, see Hjorth-Jensen [1] p. 317. However, there is one important difference compared to the 1D linear system: The matrix is not tridiagonal anymore! This means that we cannot apply the Thomas algorithm anymore, and standard methods as LU-decomposion becomes unpractical. However, in our case the matrix is positive definite, implying that iterative schemes converges to the tru solution, Hjorth-Jensen [1] p. 322. Iterative methods are often much faster than direct methods like Gaussian elimination and LU-decomposition. Hence we will solve the implicit problem applying iterative methods.

1.7.3 Iterative methods

This section follows Olver and Shakiban's [3] chapter on iterative methods closesly. The goal is to present the main idea behind iterative methods, and to present the Jacobi method as part of a general framework.

Firstly, an iterative method is not a direct method, like e.g. Gaussian elemination, that solves the problem as it is stated, typically

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \tag{36}$$

Instead, with iterative methods, one solves another problem:

$$\mathbf{u}^{k+1} = \mathbf{T}\mathbf{u}^k + \mathbf{c}, \tag{37}$$

where k stands for iterations.

For the iterative method to work, it firstly has to converge:

$$\lim_{k\to\infty} \mathbf{u}^k = \mathbf{u}^*. \tag{38}$$

Taking the limit on both sides of (37) and using (38) we get

$$\lim_{k\to\infty} \mathbf{u}^{k+1} = \lim_{k\to\infty} \mathbf{T}\mathbf{u}^k + \mathbf{c} \tag{39a}$$

$$\mathbf{u}^* = \mathbf{T}\mathbf{u}^* + \mathbf{c}. \tag{39b}$$

(39b) is a fixed point equation, and we have that the solution to the iterative scheme has to converge to a fixed point.

Secondly, the convergent solution $\mathbf{u}^*$ that is approached as $k \to \infty$ must become equal to the solution $\mathbf{u}$ of the original problem, $\mathbf{A}\mathbf{u} = \mathbf{b}$.

Lets now look at conditions for the 1st condition, convergence to the fixed point $\mathbf{u}^*$ to hold. We study the error with respect to the fixed point

$$\begin{aligned}
\mathbf{e}^{k+1} &= \mathbf{u}^{k+1} - \mathbf{u}^* && \text{(40a)}\\
&\overset{(37)}{=} \mathbf{T}\mathbf{u}^k + \mathbf{c} - \mathbf{u}^* && \text{(40b)}\\
&= \mathbf{T}\mathbf{u}^k + \mathbf{c} - (\mathbf{T}\mathbf{u}^* + \mathbf{c}) && \text{(40c)}\\
&= \mathbf{T}(\mathbf{u}^k - \mathbf{u}^* && \text{(40d)}\\
&= \mathbf{T}\mathbf{e}^k && \text{(40e)}\\
&= \mathbf{T}^k\mathbf{e}^{(0)}. && \text{(40f)}
\end{aligned}$$

(40f) shows that the development of the error in convergence towards the fixed point depends on the same matrix as the original iterative problem, $\mathbf{T}$. We have convergende towards the fixed point $\mathbf{u}^*$ if $\mathbf{T}$ is a convergent matrix. $\mathbf{T}$ is a convergent matrix if the spectral radius of $\mathbf{T}$ is less than one, $\rho(T) < 1$. The speed of convergence is inversly related to the spectral radius, implying that we want to construct T with as low spectral radius as possible.

We now have a condition for the first requirement for our iterative scheme: convergence to the fixed point. Now for the 2nd condition: convergence to the true solution. An iterative scheme converging to a fixed point different from the true solution is of little use in our case. It turns out that convergence to the true solution depeneds on the chosen scheme, represented by the matrix T in our case, and the coefficient matrix of the original problem, A. We have that both the Jacobi scheme, to be derived, and the Gauss-Seidel scheme are convergent to the true solution if A is strictly diagonally dominant.

We sum up our results in this section:

The iteration scheme (37) converges to the solution of the original linear system (36) if the following two conditions are satisfied

- The spectral radius of T in (37) is less than one.
- A in (36) is strictly diagonally dominant (In the case of Jacobi and Gauss-Seidel)

Next we will derive the Jacobi-algorithm and relate it to our genereal iterative system (37).

1.7.4 Jacobi's algorithm

We begin by a rewrite of the linear system (36), where we split A into one strictly lower triangular matrix, L, one strictly upper triangular matrix, U, and one diagonal matrix, D:

$$\begin{aligned}
A\mathbf{u} &= \mathbf{b} && \text{(41a)}\\
(L + D + U)\mathbf{u} &= \mathbf{b} && \text{(41b)}\\
\mathbf{u} &= D^{-1}\Big(- (L + D)\mathbf{u} + \mathbf{b}\Big) && \text{(41c)}
\end{aligned}$$

The above is nothing new. It is still the original system. With the Jacobi algorithm, one sets $\mathbf{u}^{k+1} = \mathbf{u}$ on the left hand side of (41c) and $\mathbf{u} = \mathbf{u}^k$ on the right hand side to get

$$\mathbf{u}^{k+1} = D^{-1}\Big(- (L + D)\mathbf{u}^k + \mathbf{b}\Big) \quad \text{(42a)}$$

We see that the Jacobi-scheme (42a) fits our general iterative fixed point scheme (37), with

$$\begin{aligned}
T &= -D^{-1}(L + D) && \text{(43a)}\\
\mathbf{c} &= D^{-1}\mathbf{b}. && \text{(43b)}
\end{aligned}$$

It is perhaps easier to see that the Jacbi algoritm (42a) do by considering a single row in the matrix-vector system

$$u_i^{(k+1)} = -\frac{1}{a_{ii}} \sum_{j=1, i \neq j} a_{ij} u_j^{(k)} + \frac{b_i}{a_{ii}}. \tag{44a}$$

Using (44) we will in the next section see that it is easy to formulate a Jacobi-solver for the 2D implicit scheme (34e).

1.7.5 Jacobi's algorithm applied to the 2D implicit diffusion

Now we will apply Jacobi's algorithm (44) to the 2D implicit scheme (34e). We see that (34e) is already almost set up like a Jacobi-scheme. We get

$$u_{i,j}^{n,(k+1)} = u_{i,j}^{n-1} + \frac{1}{1+4\alpha}(u_{i-1,j}^{n,(k)} + u_{i+1,j}^{n,(k)} + u_{i,j-1}^{n,(k)} + u_{i,j+1}^{n,(k)}) \tag{45a}$$

(45) is our Jacobi-Solver. By setting

$$a_{ii} = -(1+4\alpha) \tag{46a}$$

$$b_i = a_{ii} u_{i,j}^{n-1} \tag{46b}$$

in (45) we see that (45) corresponds to the Jacobi-algorithm (44). We also note that only 4 neighboring pints are needed.

Following Hjorth-Jensen [1] p. 319, we get the following main algorithm for the Jacobi-solver

- 1: Make intial guess for $u_{i,j}$ for all internal points.
- 2: Caclulate iteration k of u, u^k using (34e).
- 3: Stop if wanted convergence reached, else continue next iteration.
- 4: Set previous u-value equal to new u.
- 5: Repeat all steps from step 2.

The above algorithm is used for all time steps, starting with the first internal time step, solving one step at the time before moving to the next step.

1.8 2D Physical problem with boundary conditions

We choose to model a laminar flow inside an infinite tunnel with a finite quadratic cross section. We only model a single cross section, the problem being identical for all cross sections because of the homogenity assumption in the flow direction.

There will be no slip boundary conditions on all four walls. The floor and the sides will be fixed, while the roof will be moving in the flow direction. With these boundary conditions we have zero velocity on the floor and the side walls, while we have a non-zero velcocity in the flow direction. The figure below shows our problem.

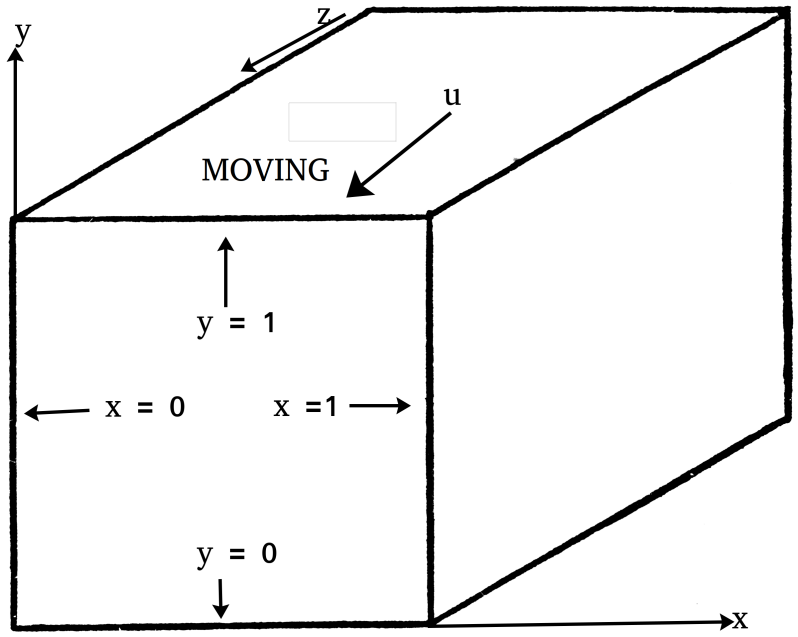

Figure 1: 2D problem sketch

With our choise of coordinate system, positive flow represents flow out of the paper.

We choose $u(x, y = 1) = 1$, and $u(x = 0, y) = u(x = 1, y) = u(x, y = 0) = 0$.

We model the flow with the heat equation. This equation can be derived from the equations of incompressible flow, the Navier-Stokes equations, by assuming laminar flow and a flow profile of type $u = u(x, y)$:

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \nabla^2 \mathbf{u} \tag{47a}$$

$$w(x, y)_t + (w\frac{\partial}{\partial z})w(x, y) \overset{\text{Zero pressure gradient}}{=} (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial x^2})w(x, y) \tag{47b}$$

$$w(x, y)_t = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial x^2})w(x, y) \tag{47c}$$

Central assumptions behind the resulting heat equation is that there is no external pressure gradient and that the flow is laminar. With turbulent flow we would have flow in all directions. This implies that our heat equation is only valid for Reynolds numbers giving laminar flow.

2 Bibliography

[1] Hjorth-Jensen, M.(2015) Computational physics. Lectures fall 2015. `https://github.com/CompPhysics/ComputationalPhysics/tree/master/doc/Lectures`

[2] `https://github.com/CompPhysics/ComputationalPhysics/tree/master/doc/pub/pde`

[3] Olver, P.J and Shakiban, C.(2006) Applied linear algebra. Pearson Prentice Hall.