GEF4510 PROBLEM SET 06

Jostein Brændshøi

October 30, 2016

Flux correction methods

a) We first note the advective flux given in the exercise by

$$F_j^n = \frac{1}{2}\left[\left(u_j^n + |u_j^n|\right)\theta_j^n + \left(u_{j+1}^n - |u_{j+1}^n|\right)\theta_{j+1}^n\right]\frac{\Delta t}{\Delta x} \tag{1}$$

We start by explicitly writing out the expressions for F_j^n for both when u_j^n is positive and when it's negative:

$$u_j^n \geq 0 \qquad \Rightarrow \qquad F_j^n = \frac{1}{2}\left[2u_j^n\theta_j^n\right]\frac{\Delta t}{\Delta x} = u_j^n\frac{\Delta t}{\Delta x}\theta_j^n \tag{2}$$

$$u_j^n < 0 \qquad \Rightarrow \qquad F_j^n = \frac{1}{2}\left[-2|u_{j+1}^n|\theta_{j+1}^n\right]\frac{\Delta t}{\Delta x} = -|u_{j+1}^n|\frac{\Delta t}{\Delta x}\theta_{j+1}^n \tag{3}$$

Then we remind ourselves with the given scheme

$$\theta_j^{n+1} = \theta_j^n - \left(F_j^n - F_{j-1}^n\right) \tag{4}$$

and furthermore insert (2) and (3) into (4). Let's start with (2). This gives

$$\theta_j^{n+1} = \theta_j^n - \left(u_j^n\frac{\Delta t}{\Delta x}\theta_j^n - u_{j-1}^n\frac{\Delta t}{\Delta x}\theta_{j-1}^n\right) = \theta_j^n - \frac{\Delta t}{\Delta x}\left(u_j^n\theta_j^n - u_{j-1}^n\theta_{j-1}^n\right)$$

$$\Rightarrow \qquad \frac{\theta_j^{n+1} - \theta_j^n}{\Delta t} = -\frac{u_j^n\theta_j^n - u_{j-1}^n\theta_{j-1}^n}{\Delta x}$$

where the last implication is done through rearrangment of the above equation. Here we more clearly see that this is a discretization of the advection equation (flux form). We recognice the left-hand-side to be a one-sided forward FDA to $\partial\theta/\partial t$ and we know from Taylor series that this FDA is first order accurate in time since we truncate all terms of Δt and higher. On the right-hand-side we recognice a one-sided backwards FDA for $\partial A/\partial x,\ A = u\theta$. Again, from Taylor series we know that this FDA is first order accurate in space. Combining these two results gives us that the scheme (4) is first order in time and space when $u_j^n \geq 0$. Now to the case $u_j^n < 0$. Inserting (3) into (4) yields

$$\theta_j^{n+1} = \theta_j^n - \left[-|u_{j+1}^n|\frac{\Delta t}{\Delta x}\theta_{j+1}^n + |u_j^n|\frac{\Delta t}{\Delta x}\theta_j^n\right] = \theta_j^n + \frac{\Delta t}{\Delta x}\left(|u_{j+1}^n|\theta_{j+1}^n - |u_j|^n\theta_j^n\right)$$

$$\Rightarrow \qquad \frac{\theta_j^{n+1} - \theta_j^n}{\Delta t} = \frac{|u_{j+1}^n|\theta_{j+1}^n - |u_j|^n\theta_j^n}{\Delta x}$$

where we see that the sign on the right-hand-side is opposite in this case. This is because $u_j^n < 0$. Here the left-hand-side is as before and the right-hand-side is now a forward difference instead of backwards, but this is also accurate to first order. So, by the same argument as for the case above, the scheme is first order in time and space also when $u_j^n < 0$.

b) For the stability analysis first look at the case $u_j^n > 0$ and employ Von Neumann's method which starts by define a Discrete Fourier component for the solution

$$\theta_j^n = \Theta_n e^{i\alpha j\Delta x}$$

and the inserting this into our scheme (for $u_j^n > 0$) to find the growth factor G

$$\Theta_{n+1}e^{i\alpha j\Delta x} = \Theta_n e^{i\alpha j\Delta x} - \frac{\Delta t}{\Delta x}\left(u_j^n\Theta_n e^{i\alpha j\Delta x} - u_{j-1}^n\Theta_n e^{i\alpha(j-1)\Delta x}\right)$$

$$\Rightarrow \qquad G = 1 - \frac{\Delta t}{\Delta x}\left(u_j^n - u_{j-1}^n e^{-i\alpha\Delta x}\right) = 1 - C_j^n + C_{j-1}^n\cos(\alpha\Delta x) - iC_{j-1}^n\sin(\alpha\Delta x)$$

Here the resulting equation has been solved for $G = \Theta_{n+1}/\Theta_n$ where $C_j^n = u_j^n\Delta t/\Delta x$ is the Courant number. Next we investigate the absolute value $|G|$ of the growth factor:

$$\begin{aligned}
|G| &= \sqrt{\left[1 - C_j^n + C_{j-1}^n\cos(\alpha\Delta x)\right]^2 + \left[C_{j-1}^n\sin(\alpha\Delta x)\right]^2} \\
&= \sqrt{1 - 2C_j^n + (C_j^n)^2 + (C_{j-1}^n)^2 + 2C_{j-1}^n\cos(\alpha\Delta x)(1 - C_j^n)} \\
&= \sqrt{(1 - C_j^n)^2 + (C_{j-1}^n)^2 + 2C_{j-1}^n\cos(\alpha\Delta x)(1 - C_j^n) + 1 - 1} \\
&= \sqrt{1 + (1 - C_j^n)^2 - (1 - C_{j-1}^n)(1 + C_{j-1}^n) + 2C_{j-1}^n(1 - C_j^n)\cos(\alpha\Delta x)}
\end{aligned}$$

From the first to the second step we have multiplied out the parenthesis and in the third step we add and subtract one to rearrange our expression. When people implement scheme (4) on a computer it is ussually with a substantial amount of gridpoints and therefore with Δx that's not too huge. It would therefore seem reasonable to make the assumption that the advection speed at grid point $j-1$ is almost the same as the advection speed at point j, or mathematically that $u_{j-1}^n \approx u_j^n$, which implies $C_{j-1}^n \approx C_j^n$. Using this we can make further calculations on $|G|$ to get

$$\begin{aligned}
|G| &= \sqrt{1 + (1 - C_j^n)\left[(1 - C_j^n) - (1 + C_j^n) + 2C_j^n\cos(\alpha\Delta x)\right]} \\
&= \sqrt{1 + (1 - C_j^n)\left[-2C_j^n + 2C_j^n\cos(\alpha\Delta x)\right]} \\
&= \sqrt{1 - 2C_j^n(1 - C_j^n)\left[1 - \cos(\alpha\Delta x)\right]}
\end{aligned}$$

where we see that the factors $2C_j^n$ and $1 - \cos(\alpha\Delta x)$ are both always positive. To keep $|G| \leq 1$ for all j, n we see that we must require the maximum value of C_j^n to be less than 1. This gives the stability condition

$$|G| \leq 1 \qquad \Rightarrow \qquad \max_{j,n}\{C_j^n\} = \max_{j,n}\left\{u_j^n\frac{\Delta t}{\Delta x}\right\} \leq 1$$

Then to the case $u_j^n < 0$. Inserting the discrete Fourier component yields the growth factor

$$\Theta_{n+1}e^{i\alpha j\Delta x} = \Theta_n e^{i\alpha j\Delta x} + \frac{\Delta t}{\Delta x}\left(|u_{j+1}^n|\Theta_n e^{i\alpha(j+1)\Delta x} - |u_j^n|\Theta_n e^{i\alpha j\Delta x}\right)$$

$$\Rightarrow \qquad G = 1 + \frac{\Delta t}{\Delta x}\left(|u_{j+1}^n|e^{i\alpha\Delta x} - |u_j^n|\right) = 1 - C_j^n + C_{j+1}^n\cos(\alpha\Delta x) + iC_{j+1}^n\sin(\alpha\Delta x)$$

and when we make the assumption $C_{j+1}^n \approx C_j^n$ we get the same expression under the square root for $|G|$ (where $|u_j^n|$ is the same as u_j^n was before) meaning the subsequent computations will be the same as before and hence the scheme is stable under the same condition in this case too.

c) To find the numerical diffusion our scheme, we first substitute (1) into (4) to obtain

$$\begin{aligned}\theta_j^{n+1} &= \theta_j^n - \frac{\Delta t}{2\Delta x}\left[(u_0+|u_0|)\theta_j^n + (u_0-|u_0|)\theta_{j+1}^n - (u_0+|u_0|)\theta_{j-1}^n + (u_0-|u_0|)\theta_j^n\right] \\ &= \theta_j^n - \frac{\Delta t}{2\Delta x}\left[2|u_0|\theta_j^n + (u_0-|u_0|)\theta_{j+1} - (u_0+|u_0|)\theta_{j-1}^n\right]\end{aligned}$$

and then we recall (explictly to second order) the Taylor series for the terms θ_j^{n+1} and $\theta_{j\pm1}^n$

$$\begin{aligned}\theta_j^{n+1} &= \theta_j^n + \left.\frac{\partial\theta}{\partial t}\right|_j^n \Delta t + \frac{1}{2}\left.\frac{\partial^2\theta}{\partial t^2}\right|_j^n \Delta t^2 + \mathcal{O}(\Delta t^3) \\ \theta_{j\pm1}^n &= \theta_j^n \pm \left.\frac{\partial\theta}{\partial x}\right|_j^n \Delta x + \frac{1}{2}\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta x^2 + \mathcal{O}(\Delta x^3)\end{aligned}$$

and furthermore insert these into our expression for the scheme while omitting the order terms $\mathcal{O}(\Delta t^3)$, $\mathcal{O}(\Delta x^3)$ for readability (this means approximation to second order). Doing this gives

$$\begin{aligned}\theta_j^n + \left.\frac{\partial\theta}{\partial t}\right|_j^n \Delta t + \frac{1}{2}\left.\frac{\partial^2\theta}{\partial t^2}\right|_j^n \Delta t^2 &= \theta_j^n - \frac{\Delta t}{2\Delta x}\Bigg\{2|u_0|\theta_j^n] + (u_0+|u_0|)\left[\theta_j^n + \left.\frac{\partial\theta}{\partial x}\right|_j^n \Delta x + \frac{1}{2}\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta x^2\right] \\ &\qquad -(u_0-|u_0|)\left[\theta_j^n - \left.\frac{\partial\theta}{\partial x}\right|_j^n \Delta x + \frac{1}{2}\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta x^2\right]\Bigg\} \\ &= \theta_j^n - \frac{\Delta t}{2\Delta x}\left(2u_0\left.\frac{\partial\theta}{\partial x}\right|_j^n \Delta x - |u_0|\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta x^2\right) \qquad (5)\end{aligned}$$

If we now recall the original advection equation, we can write the following implication when differentiating with time and substituting the advection equation into the differentiated expression:

$$\frac{\partial\theta}{\partial t} = -u_0\frac{\partial\theta}{\partial x} \quad \Rightarrow \quad \frac{\partial^2\theta}{\partial t^2} = -\frac{\partial}{\partial t}\left(u_0\frac{\partial\theta}{\partial x}\right) = \frac{\partial}{\partial x}\left(u_0\frac{\partial\theta}{\partial t}\right) = -\frac{\partial}{\partial x}\left(-u_0^2\frac{\partial\theta}{\partial x}\right) = u_0^2\frac{\partial^2\theta}{\partial x^2}$$

We can now insert the result found here for $\partial^2\theta/\partial t^2$ into (5). We start by rearrangning (5) and collect terms in the advection equation on the left-hand-side and divide by Δt , and then we insert:

$$\begin{aligned}\left.\frac{\partial\theta}{\partial t}\right|_j^n + u_0\left.\frac{\partial\theta}{\partial x}\right|_j^n &= \frac{1}{2}|u_0|\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta x - \frac{1}{2}\frac{\partial^2\theta}{\partial t^2}\Delta t \\ &= \frac{1}{2}|u_0|\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta x - \frac{1}{2}u_0^2\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \Delta t \\ &= \frac{1}{2}\left(|u_0|\Delta x - u_0^2\Delta t\right)\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n \\ &= \frac{1}{2}|u_0|\left(\Delta x - |u_0|\Delta t\right)\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n = \kappa^*\left.\frac{\partial^2\theta}{\partial x^2}\right|_j^n\end{aligned}$$

Here we see that after using the Taylor series we get back the advection equation (confirming a consistent scheme), but as part of the higher order error terms we see a diffusion-like term with diffusion coefficient $\kappa^* = (1/2)|u_0|\left(\Delta x - |u_0|\Delta t\right)$. This term represents the numerical diffusion of the scheme (4).

d) When implementning the scheme (4) we can make use of the fact that we are dealing with the case $u_j^n = u_0 > 0$. Based on the computations done in **a)** this gives us the scheme

$$\theta_j^{n+1} = \theta_j^n - \frac{\Delta t}{\Delta x}\left(u_j^n \theta_j^n - u_{j-1}^n \theta_{j-1}^n\right) = \theta_j^n - C\left(\theta_j^n - \theta_{j-1}^n\right) \tag{6}$$

So here (4) is reduced to the standard upwind scheme for $u_0 > 0$. But when doing the numerical computations we also need to know for how many spatial and temporal gridpoints we should the simulation. Regarding space we have x_j at the j'th gridpoint given by $x_j = (j-1)\Delta x$. This gives us the total number of gridpoints j_m at $x_j = L$ by

$$(j_m - 1)\Delta x = L \qquad \Rightarrow \qquad j_m = \frac{L}{\Delta x} + 1 = \frac{10L}{\sigma} + 1 = \frac{10 \cdot 10L}{L} + 1 = 101 \tag{7}$$

where we have used the given definitions of σ and Δx. Next we find the required number of time points in order to simulate the system for the requested number of cycles (25). We note that the time t_n required for a fluid parcel to be advected from a given space point in one cycle to the same space point in the next cycle with an advection speed u_0 is $t_n = n\Delta t = L/u_0$. This means the total number of required time points n_m for a fluidparcel to be advected N cycles is given by

$$n_N \Delta t = \frac{NL}{u_0} \qquad \Rightarrow \qquad n_N = \frac{NL}{u_0 \Delta t} \tag{8}$$

so in other words; N times the number of time points required for one cycle. Both the results (7) and (8) are used in the code for implementing (6) and for writing θ, at the correct n-values corresponding to the requested cycles, to file for plotting. Also worth mentioning is the implementation of the boundary condition. For it to be cyclic we require θ at the first space to be equal to its value at the final space point, i.e. $\theta(0,t) = \theta(L,t),\ \forall\ t$.

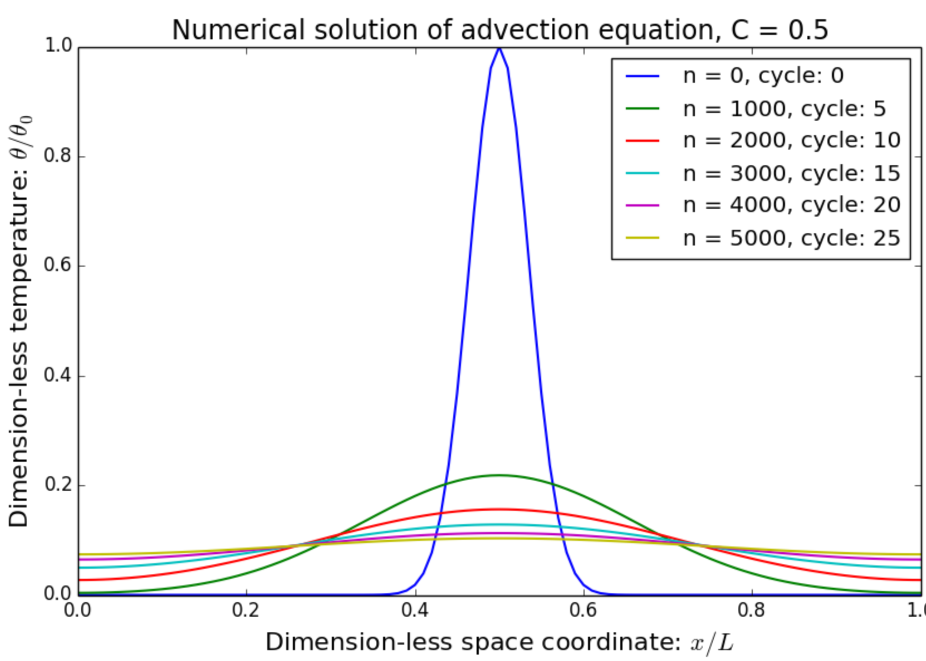

Figure 1: Solution to advection equation ($C = 0.50$) with Gaussian bell IC and cyclic BC.

All of the above discussed numerics together with the given imformation from the exercise are implemented in the Fortran file `advection_solver.f90`. This file also contains code for the exercise **e)** and **g)**. When running the program together with visualization script `visualize_results.py` we get the plot diaplyed in Figure 1. Here we see that the initial distribution is advected and that the general form of the initial Gaussian bell is still very much recognisable. However we see a significant smoothing process in which the slope of the bell is lessend with time and the values for θ are quite offset at later times. This is caused by the numerical diffusion inherent in our upwind scheme (6) and it really looks like both an advective and diffusive process is present in our physical system although the latter is only artificial and not representative for what would happen in the real world. So one goal would be to dampen this numerical diffusion effect. This brings us to the next exercise.

e) Based on the MPDATA method we can make a flux correction to the numerical diffusion seen in **d)**. Using the description of the algorithm from the syllabus PDF, we can write the code found under the if-statement `if (method_type == "mpdata") then`. Here we execute the MPDATA for `num_iterations` when the iterative approach is desired (specified by `use_scaling = .FALSE.`). When running the code with `num_iterations = 1` and `num_iterations = 2` we get the result in the left and right panel (respectively) of Figure 2.

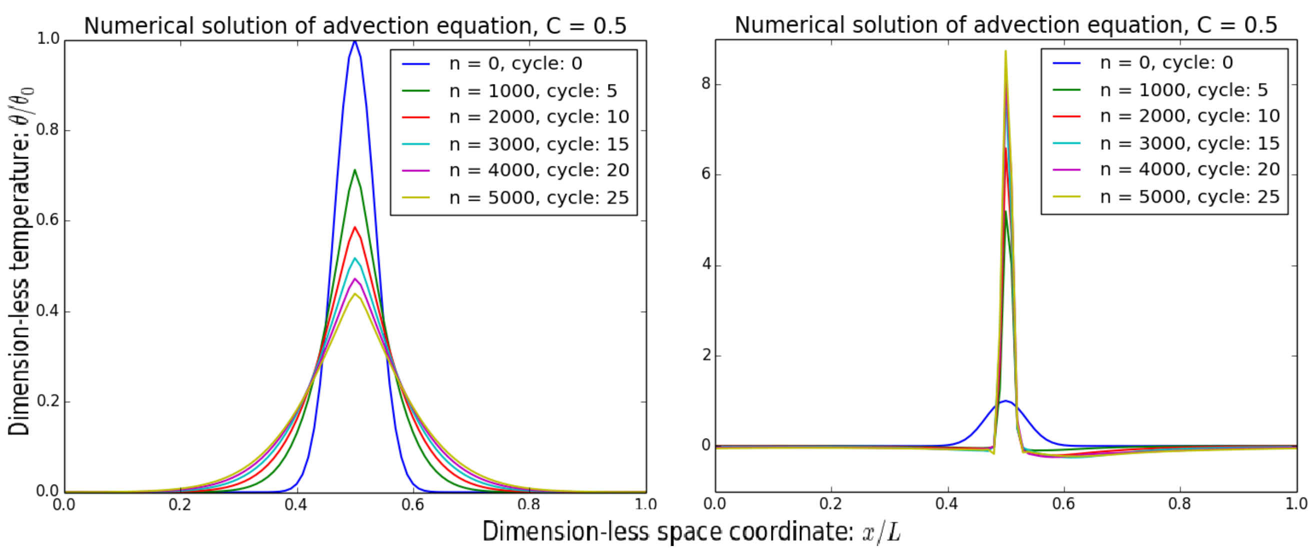

Figure 2: MPDATA with one iteration (left) and two iterations (right)

We see that the flux correction seem to have worked well for the case with one iteration. Here we clearly see an improvement compared to the case in **d)** where no flux correction was used. So the anti-diffusive velocity u_j^* have resteepend the slopes of the Gaussian bell, but not quite enough so that the original shape is preserved in time. When using two iterations we would expect a more accurate result because (in theory) the numerical diffusion present in the corrector step would also be taken care of. However, unfortunately, the iterative method does not seem to be working for more than one iteration. We get very large values which does not make physical sense and is definetly not in line with what we would expect to see. And using even more iterations, the θ-values become even larger. So there is likely an issue with how I have implemented the iterative method, but this issue was sadly not found during testing.

Next we turn our attention to the scaling method. A plot of the result with $S_c = 1.1$ is given in Figure 3. It was tried with $S_c = 1.3$, but this gave too large θ-values so 1.1 was used instead.

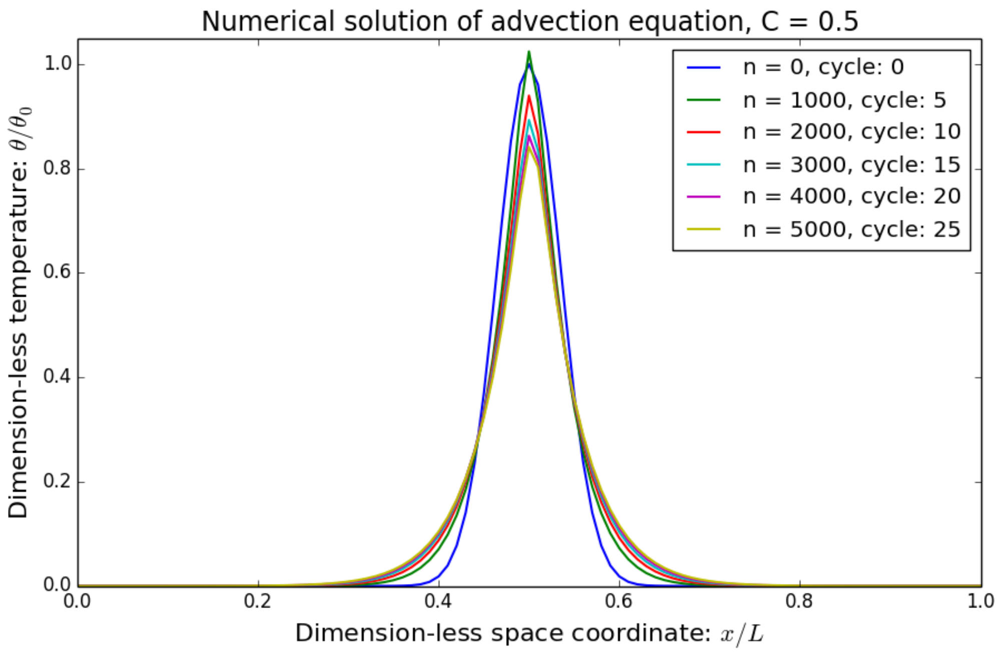

Figure 3: MPDATA with scaling approach ($S_c = 1.1$)

Here we see that the scaling approach seem to do the job quite nicely. Even though we still get some numerical diffusion, the initial bell is preserved reasonably well. It gives a far better result than the single iteration in Figure 2 at the cost of only one more floating-point-operation per space-time point. However, for more iterations, we would expect the iterative method to work better as this, in theory, would perfectly cancel all numerical diffusion (as number of iterations goes to infinity), but this is again a more computationally heavy task than the simple scaling method. Also worth mentioning is that the iterative metod has more of a theoretical justification behind it, whilst the scaling method is more based in trial and error and thus perhaps the less elegant approach from an aesthetic point of view.

f) We add the small number ϵ in the denominator to handle the possible case in which both $\theta^n_{j+1} - \theta^n_{j-1}$ and θ^n_j are zero at the same time. One could imagine this might happen in the tails of the bell shape as the slope of θ here is basically zero as well as θ itself is very near zero. Then by adding ϵ we avoid ending up with and undefined expression $0/0$ which cannot be handled by the computer. So in other words; where both the mentioned terms are zero we make sure the term $(1/\theta)\partial\theta/\partial x$, and hence u^*_j, is zero.

g) As hinted at in **e)**, when using $S_c > 1.1$ we get too large θ-values as a consequense of S_c over compensating with the anti-diffusive velocity u^*_j and giving too steep slopes resulting in narrower and taller bell than the initial condition. Choosing $S_c < 1.1$ (but keeping it above 1) gives a better result than panel left in Fig. 2, but not as good as what seems to be the optimal value for S_c, namely 1.1. Regarding the possible choices of FDA when computing u^*_j, there is, in the code, added support for choice between a centered, forward and backward FDA for $\partial\theta^*/\partial x$. In figure 4 we can see the scaled single-iteration approach (similar to that found in of Fig. 3) with a forward and backward FDA.

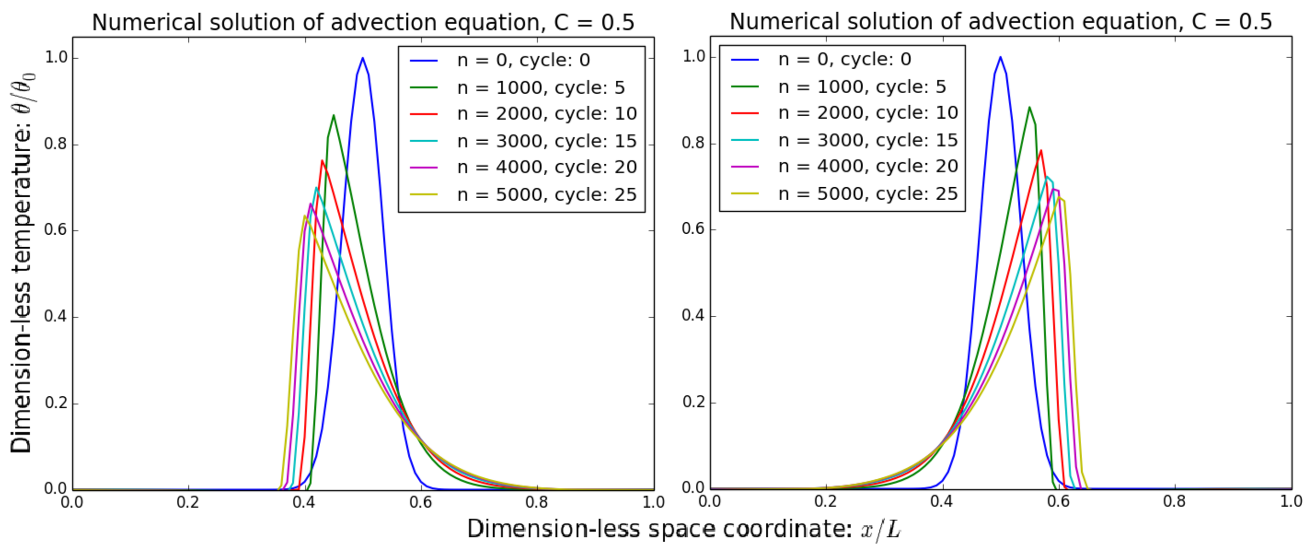

Figure 4: MPDATA with scaling approach ($S_c = 1.1$) and forward FDA (left) and backward (right) when computing the anti-diffusive velocity u_j^*.

When looking at the plot there are three features that stand out. Firstly we notice how the slope on the left and right side of the peak, for the forward and backward FDA respectively, are very steep compared to the other side of the peak. This asymmetri seem to be directly linked to the two FDA's. We also note how the scaling factor $S_c = 1.1$ did not make as much of a difference here as it did for the centered FDA case. When using a forward and backward approximation it seems that we would need a larger scaling factor. After some testing $S_c = 1.2$ seem to be a better choice for these cases.

Finally we note that that the peak of the bell shape seem be shifted to the left when using a forward FDA and to the right when using a backward FDA. This is in contrast to the centered non-shifted case when using the centered FDA in **e)**. The reason for this shift might be explained by considering the value of u_j^* at the peak. When we used the centered FDA our nominator contains $\theta_{j+1}^* - \theta_{j-1}^*$, meaning that when we are at j corresponding to the peak location, we subtract two, hopefully, equal θ^* values resulting in a zero-valued anti-diffusive velocity. This furthermore means the peak is advected with the intended advection velocity u_0 (there is no numerical diffusion at the peak since $\partial\theta/\partial x = 0$ here). But this is not the case when we don't use a centered FDA. For example when using a forward FDA we compute $\theta_{j+1}^* - \theta_j^*$ and when we are at the peak this means subtracting the peak point value from a point past (to the right) the peak, resulting in a negative value for u_j^*. Furthermore this means making the net advection speed $u_0 + u^*$ less than what it physically should be (u_0) at the peak. This again results in the peaking reaching a shorter distance in the same time period which is manifested (in the plot) through the fact that the peak does not reach back to its original position at $L/2$. We also see that the longer time that has passed, the larger the shift is, which makes intuitively sense. The excact same argument goes for the backward FDA, but here we instead get too large ($u_j^* > 0$) net advection velocity for the peak due to using a point before (to the left) the peak, resulting in the peak reaching a longer distance in the same time period and thus giving a right-shifted peak.