Numerical solution of differential equations
Project 2

Lars A. M. Olsen, Ida M. Sandum, Markus A. Stokkenes

Introduction

In this project we analyse some numerical methods for solving PDEs. We develop and implement some finite difference schemes for elliptic 2D-problems with complex domains so that the a regular grid is no longer available. We also do this for 1D hyperbolic problems with variable coefficients. The analysis includes theoretical error estimates and pinpointing and resolving possible deficiencies of the schemes.

Part 1: Heat distribution in anisotropic materials

We are given the PDE

$$-a\partial_x^2 u - (\vec{d_2}\cdot\nabla)^2 u = f,$$

where $\vec{d_2} = (1, r)$ and $a > 0$. The domain is the unit square $\Omega = [0,1]^2$, and we have Dirichlet boundary conditions $u = g$ on $\partial\Omega$. We discretize the equation using second order central differences in the directions $\vec{d_2}$ and $\vec{d_1} = (0,1)$. Noticing that

$$(\vec{d_2}\cdot\nabla)^2 u = ||\vec{d_2}||_2^2 (\frac{\vec{d_2}}{||\vec{d_2}||_2}\cdot\nabla)^2 u,$$

the PDE can be rewritten as:

$$-a\partial_x^2 u - (1+r^2)\partial_{d_2}^2 u = f,$$

where $\partial_{d_2}^2 u$ denotes the second directional derivative in the $\vec{d_2}$ direction. If we let $r = 1$, the equation becomes

$$-a\partial_x^2 u - 2\partial_{d_2}^2 u = f.$$

To discretize this equation, we first partition the x-axis into M intervals, and let the horizontal step size be $h := 1/M$. Since $r = 1$ we have that $\vec{d_2} = (1,1)$, which means that the vertical step size will also equal h, and the step size in the $\vec{d_2}$ will be $\sqrt{2}h$. Using central differences to approximate the derivatives, we obtain the scheme

$$-\mathcal{L}_h U_m^n = f_m^n, \qquad m, n = 0, \dots, M$$

where $\mathcal{L}_h = \frac{1}{h^2} a\delta_x^2 + \frac{2}{2h^2}\delta_{d_2}^2 = \frac{1}{h^2}(a\delta_x^2 + \delta_{d_2}^2)$. Expanding this out yields

$$\frac{1}{h^2}\Big((2a+2)U_m^n - aU_{m+1}^n - aU_{m-1}^n - U_{m+1}^{n+1} - U_{m-1}^{n-1}\Big) = f_m^n.$$

Implementation

Before we can implement the scheme, we need to choose an ordering of the nodes in the grid. We chose to start in the bottom left corner of the grid and then order going from left to right. When we reach a boundary point in $x = 1$, we jump back to $x = 0$ and thus always go from left to right. An example of the ordering for 9 grid points, using 0-indexing because it is easier to transform into python-code:

$$\begin{bmatrix} 6 & 7 & 8 \\ 3 & 4 & 5 \\ 0 & 1 & 2 \end{bmatrix}$$

We can write the scheme as a linear system $A\vec{U} = \vec{b} + \vec{F}$, where $\vec{b}$ is a vector containing the boundaries and $\vec{F}$ is a vector containing information from the right hand side f. This gives a $(M+1)^2 \times (M+1)^2$ matrix, where $M+1$ is the number of grid points on the x-axis.

The matrix A has 1 on the diagonal and 0 elsewhere for the first and last M+1 elements. These account for the boundaries in $y = 0$ and $y = 1$. Then the following pattern repeats itself $M-1$ times: one row with 1

on the diagonal and zero elsewhere, $M-1$ rows with $-a$ on the sub- and superdiagonal and $2a+2$ on the diagonal, as well as -1 at indices $i-M-2$ and $i+M+2$, where i is the index at the diagonal. And one more row with 1 on the diagonal and zero elsewhere. The first and last row in this pattern accounts for the boundaries in $x=0$ and $x=1$, and the rest of the rows describe the internal grid points.

In order to implement it we notice that grid element (m,n) (using our ordering of the nodes) can be written with matrix index as $(i \mod (M+1), i//(M+1))$. We can use this to easily check if we are at the boundary, by checking if $i \mod (M+1) = 0$ or $i \mod (M+1) = M$. If this is the case, we add 1 to the matrix at these indices. The internal grid points are straightforward to implement, and are set whenever we are not on the boundary. In addition, the first M+1 and last M+1 points are on the boundary, which is added as ones on the diagonal of the matrix.
Whenever we fill in the ones on the diagonal, we also fill in the boundary vector $\vec{b}$ with $g(x_m, y_n)$, where the point (x_m, y_n) correspond to the boundary grid point we are at. When we fill in the pattern at the internal grid points, we also fill in the right hand side vector $\vec{F}$ in the same way. Then we solve the linear system, and reshape $\vec{U}$ into a matrix in order to easily plot it in 3D.

In figure 1 below we have tested the scheme using two different functions, and we also plot the exact solution for comparison.

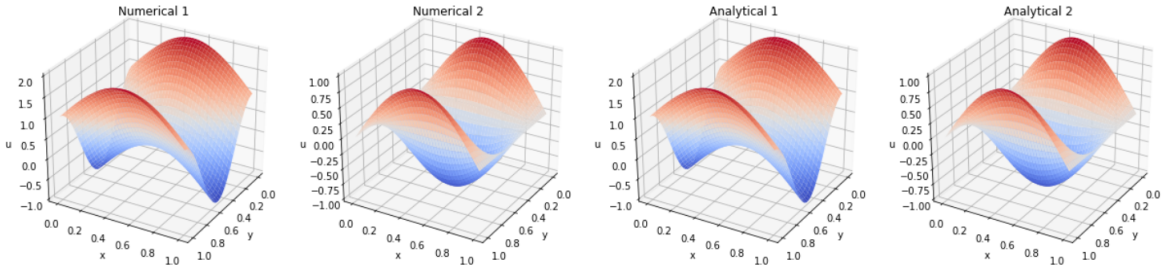


Figure 1: Numerical and analytical solutions of $u_1(x,y) = \sin(\pi x) + \cos(2\pi y)$ and $u_2(x,y) = \sin(\pi x)\cos(2\pi y)$

In figure 2 we can see that the convergence rate for the scheme is of order 2. We can verify this by looking at the error bound, which is calculated in the section "Convergence rate and error bound".

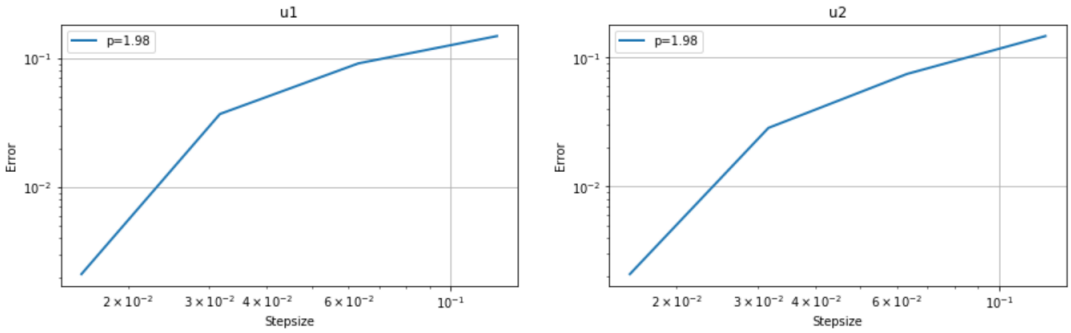


Figure 2: Convergence rate plots for $u_1(x,y) = \sin(\pi x) + \cos(2\pi y)$ and $u_2(x,y) = \sin(\pi x)\cos(2\pi y)$

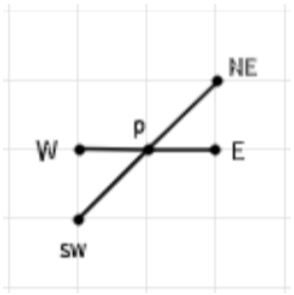
Monotonicity and stability

To show that the scheme is monotone we notice that all coefficients are positive as $h, a > 0$ We also have that $\frac{2a+2}{h^2} = \frac{a}{h^2} + \frac{a}{h^2} + \frac{1}{h^2} + \frac{1}{h^2} > 0$. This implies monotonicity.

The stencil of an interior grid point U_P is

$$\{U_P - h\vec{d_1},\ U_P,\ U_P + h\vec{d_1},\ U_P - \sqrt{2}h\vec{d_2},\ U_P + \sqrt{2}h\vec{d_2}\}.$$

This stencil can also be illustrated as



Since the scheme is monotone, we know that the discrete maximum principle for elliptic PDEs holds. It is as follows:

$$-\mathcal{L}_h U_m^n \leq 0 \implies \max_{\mathbb{G}} U_m^n \leq \max_{\partial\mathbb{G}} U_m^n,$$

where $\mathbb{G}$ is the set of all nodes, and $\partial\mathbb{G}$ is the set of all boundary nodes. We can use this to show L^∞-stability. Consider the problem $-\mathcal{L}_h V_\mathrm{P} = g_\mathrm{P}$ in $\mathbb{G}$, where $V_\mathrm{P} = 0$ in $\partial\mathbb{G}$. We introduce $W_\mathrm{P} = V_\mathrm{P} - ||\vec{g}||_\infty \phi_\mathrm{P}$, where $\phi = \frac{1}{2}x(1-x)$. Applying the scheme to W_p yields

$$\begin{aligned} -\mathcal{L}_h W_\mathrm{P} &= -\mathcal{L}_h V_\mathrm{P} + ||\vec{g}||_\infty \mathcal{L}_h \phi_\mathrm{P} \\ &= g_\mathrm{P} + ||\vec{g}||(-1) \\ &= g_\mathrm{P} - ||\vec{g}||_\infty \\ &\leq 0. \end{aligned}$$

Here we have Taylor expanded ϕ in all the desired directions and used that

$$\begin{aligned} \mathcal{L}_h \phi_\mathrm{P} &= \frac{1}{h^2}((2a+2)\phi_\mathrm{P} - a\phi_\mathrm{E} - a\phi_\mathrm{W} - \phi_\mathrm{NE} - \phi_\mathrm{SW}) \\ &= \frac{1}{h^2}((2a+2)\phi_\mathrm{P} - 2a\phi_\mathrm{P} - 2\phi_\mathrm{P} - h^2\partial_x^2\phi_\mathrm{P}) \\ &= -1 \end{aligned}$$

The discrete maximum principle then yields

$$W_\mathrm{P} \leq \max_{\partial\mathbb{G}} W_\mathrm{P} \leq 0,$$

This gives us $V_\mathrm{P} \leq ||\vec{g}||_\infty \phi_\mathrm{P} \leq ||\vec{\phi}||_\infty ||\vec{g}||_\infty$. Using the transformation $(V, g) \mapsto (-V, -g)$ and repeating the steps above, we also have $-V_\mathrm{P} \leq ||\vec{\phi}||_\infty ||\vec{g}||_\infty$. The maximum of ϕ is attained at $x = \frac{1}{2}$, and the value is $\phi(\frac{1}{2}) = \frac{1}{8}$ Thus $\|\vec{V_\mathrm{P}}\| \leq \frac{1}{8}\|\vec{g}\|_\infty$, which implies L_∞-stability for our scheme.

Convergence rate and error bound

Using the scheme on $e_\mathrm{P} = u_\mathrm{P} - U_\mathrm{P}$ yields

$$\begin{aligned} -\mathcal{L}_h e_\mathrm{P} &= -\mathcal{L}_h u_\mathrm{P} + \mathcal{L}_h U_\mathrm{P} \\ &= -\mathcal{L}_h u_\mathrm{P} - f_\mathrm{P} = -\tau_\mathrm{P} \end{aligned}$$

at the internal grid points, and $e_\mathrm{P} = 0$ at the boundary. From stability we have that

$$||\vec{e}||_\infty \leq \frac{1}{8}||\vec{\tau}||_\infty$$

.

To bound the error we therefore need to bound the truncation error. The local truncation error can be written as

$$\tau_m^n = -(\frac{a}{h^2}\delta_x^2 + \frac{1}{l^2}\delta_{d_2}^2)u_m^n - (-a\partial_x^2 u - (\vec{d_2}\cdot\nabla)^2 u_m^n).$$

Using Taylor expansions we can approximate the terms: $-\frac{a}{h^2}(U_{m+1}^n - 2U_m^n + U_{m-1}^n) \approx -\frac{a}{h^2}(h^2\partial_x^2 u_m^n + \frac{h^4}{12}\partial_x^4 u_m^n) + O(h^4)$ and $-\frac{1}{h^2}(U_{m-1}^{n-1} - 2U_m^n + U_{m+1}^{n+1}) \approx -\frac{1}{h^2}(2h^2\partial_{d_2}^2 u_m^n + \frac{1}{3}h^4\partial_{d_2}^4 u_m^n) + O(h^4)$. We thus get

$$\begin{aligned} \tau_m^n &= -\frac{a}{h^2}(h^2\partial_x^2 u_m^n + \frac{h^4}{12}\partial_x^4 u_m^n) - \frac{1}{h^2}(2h^2\partial_{d_2}^2 u_m^n + \frac{1}{3}h^4\partial_{d_2}^4 u_m^n) + a\partial_{d_2}^2 u_m^n + 2\partial_{d_2}^2 u_m^n + O(h^4) \\ &= -\frac{a}{12}h^2\partial_x^4 u_m^n - \frac{1}{3}h^2\partial_{d_2}^4 u_m^n + O(h^4) \end{aligned}$$

$\Longrightarrow$

$$|\tau_m^n| = |-\frac{a}{12}h^2\partial_x^4 u_m^n - \frac{1}{3}h^2\partial_{d_2}^4 u_m^n + O(h^4)| \leq \frac{h^2}{12}(a\max_{x,y\in\Omega}|\partial_x^4 u_m^n| + 4\max_{x,y\in\Omega}|\partial_{d_2}^4 u_m^n|)$$

We now have a bound for the error:

$$||\vec{e}||_\infty \leq \frac{1}{8}||\vec{\tau}||_\infty = ||\vec{e}||_\infty \leq \frac{1}{8}\frac{h^2}{12}(a\max_{x,y\in\Omega}|\partial_x^4 u_m^n| + 4\max_{x,y\in\Omega}|\partial_{d_2}^4 u_m^n|) = \frac{h^2}{96}(a\max_{x,y\in\Omega}|\partial_x^4 u_m^n| + 4\max_{x,y\in\Omega}|\partial_{d_2}^4 u_m^n|).$$

We see that the convergence rate is indeed of order 2 as figure 2 implies.

Irregular grid at the upper boundary

If k is equal to $|r|h$ and r is irrational, then k must also be irrational, since $h = 1/M$ is rational. Thus, since the y-values of the nodes are integer multiples of k, the y-values can never be equal to 1. In other words, it is impossible to obtain an even grid spacing if the upper boundary is at $y = 1$.

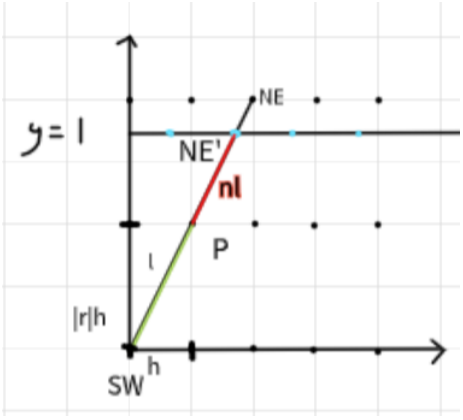


Figure 3: Irregular grid at the boundary y=1.

One way to overcome this problem is to artificially add nodes at the intersections between $y = 1$ and the lines connecting a point and its north eastern neighbor. Then we can create a modified scheme for the nodes which lie just below the boundary. Figure 3 illustrates the idea. The stencil for these points can then be written as

$$\{U_P - h\vec{d_1},\ U_P,\ U_P + h\vec{d_1},\ U_P - h\sqrt{1+r^2}\vec{d_2},\ U_P + \eta h\sqrt{1+r^2}\vec{d_2}\},$$

where $\eta \in [0, 1]$.

We use the method of undetermined coefficients to derive the modified scheme. We only need to consider the $\vec{d_2}$ direction, since the stencil in the x-direction is unchanged so that we can still use the regular central difference to approximate ∂_x^2. For the other direction, we obtain

$$\begin{aligned}
\partial_{d_2}^2 u =& c_{\mathrm{P}} u_{\mathrm{P}} + c_{\mathrm{NE'}} u_{\mathrm{NE'}} + c_{\mathrm{SW}} u_{\mathrm{SW}} - \tau \\
=& c_{\mathrm{P}} u(x_{\mathrm{P}}, y_{\mathrm{P}}) + c_{\mathrm{NE'}} u(x_{\mathrm{P}} + h, y_{\mathrm{P}} + \eta|r|h) + c_{\mathrm{SW}} u(x_{\mathrm{P}} - h, y_{\mathrm{P}} - |r|h) - \tau \\
=& c_{\mathrm{P}} u + c_{\mathrm{NE'}}(u + \eta h\sqrt{1+r^2}\partial_{\vec{d_2}} u + \frac{1}{2}\eta^2 h^2(1+r^2)\partial_{\vec{d_2}}^2 u + O(\eta^3(1+r^2)^{3/2}) + c_{\mathrm{SW}}(u - \sqrt{1+r^2}\partial_{\vec{d_2}} \\
&+ \frac{1}{2}(1+r^2)\partial_{\vec{d_2}}^2 u + O((1+r^2)^{3/2}) - \tau
\end{aligned}$$

Here τ is the truncation error and $\partial_{\vec{d_2}}$ and $\partial_{\vec{d_2}}^2$ denote the first and second directional derivatives in direction $\vec{d_2}$.
This gives the linear system

$$\begin{cases}
c_{\mathrm{P}} + c_{\mathrm{NE'}} + c_{\mathrm{SW}} = 0 \\
-\eta c_{\mathrm{NE'}} + c_{\mathrm{SW}} = 0 \\
\frac{1}{2}\eta^2(1+r^2)h^2 c_{\mathrm{NE'}} + \frac{1}{2}(1+r^2)h^2 c_{\mathrm{SW}} = 1
\end{cases}$$

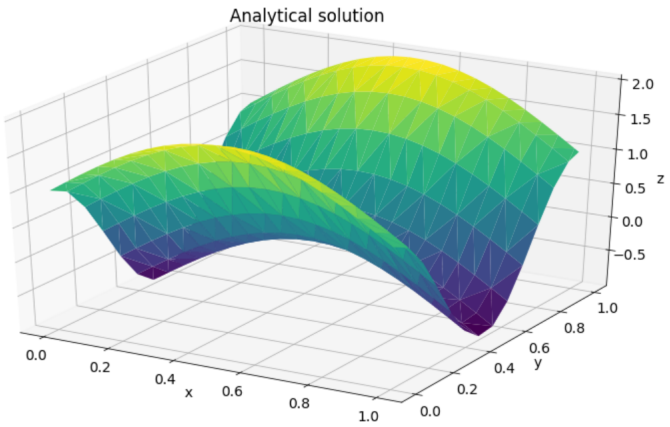
Solving this yields the coefficients

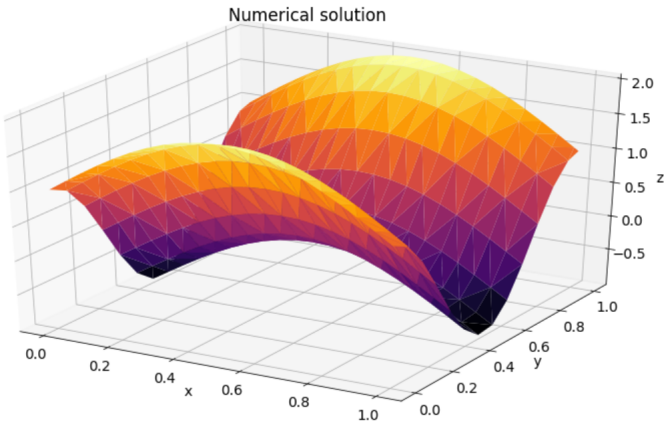
$$c_{\mathrm{P}} = \frac{-2}{h^2}\frac{1}{(1+r^2)\eta} \qquad c_{\mathrm{SW}} = \frac{2}{h^2}\frac{1}{(1+r^2)(1+\eta)} \qquad c_{\mathrm{NE'}} = \frac{2}{h^2}\frac{1}{(1+r^2)\eta(1+\eta)}$$

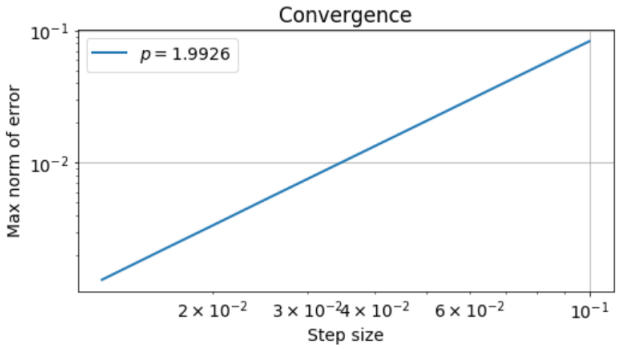
In x-direction we get the same scheme as before. Combining the two, we now have a scheme for the irregular grid in $y = 1$. The $(1 + r^2)$ factor ends up cancelling with the same factor from the PDE, and the scheme becomes as follows:

$$\frac{1}{h^2}\left((2a + \frac{2}{\eta})U_m^n - aU_{m+1}^n - aU_{m-1}^n - \frac{2}{\eta(1+\eta)}U_{m+1}^{n+1} - \frac{2}{(1+\eta)}U_{m-1}^{n-1}\right) = f_m^n$$

We have implemented this scheme in our code. For all internal nodes except those just below $y = 1$, the scheme is identical. For the special nodes, it is only a matter of calculating the diagonal distance to the points on the upper boundary, and inserting it into the scheme derived above. The implementation is not dependent on any specific ordering of the nodes. Below are plots of exact and numerical solutions for $u(x, y) = \sin(\pi x) + \cos(2\pi y)$, as well as a plot of the max error as a function of step size.





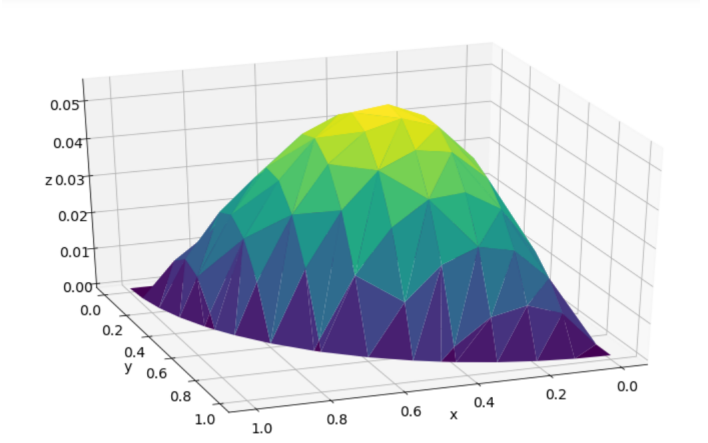


Comment: As is evident from the convergence plot, we have found the scheme to be of convergence order 2. We did not expect this beforehand, as we are dealing with an irregular grid. Intuitively, we would expect order 1, because the extra nodes are constructed from a kind of linear (first order) interpolation. We are not sure exactly why we found order 2, but are satisfied with the results nonetheless.

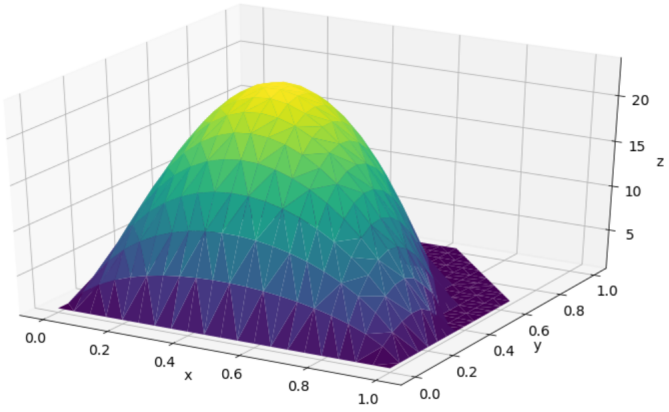
Now we turn to the case where domain is the quarter unit disk. The two choices of boundary nodes were either modifying the discretization near the boundary, or fattening the boundary. The first case is somewhat similar to the above method, where the grid is made to be denser near the boundary. This time, we have to check both the eastern and north eastern distance to the boundary. Using the method of undetermined coefficients for both the x and y directions, we obtain the following scheme:

$$\frac{1}{h^2}\Big(2(\frac{1}{\eta_1}+\frac{1}{\eta_2})U_m^n - \frac{2}{\eta_1(1+\eta_1)}U_{m+1}^n - \frac{2}{1+\eta_1}U_{m-1}^n - \frac{2}{\eta_2(1+\eta_2)}U_{m+1}^{n+1} - \frac{2}{(1+\eta_2)}U_{m-1}^{n-1}\Big) = f_m^n$$

Here, η_1 and η_2 are the fractions of the step sizes in the x and y directions, respectively. Below is a plot of the solution of $-\Delta u = 1$ using this method:



The other method is simply extending the boundary to the points outside of the quarter disk, such that the grid is equidistant. Thus we can use the plain old 5-point formula for the Laplacian. An example plot of a solution to the same equation as above is plotted below, $M = 20$ and boundary of thickness 0.2:



Comments: Concerning which scheme is faster and easier, we would definitely say fattening the boundary, as it allows for a simpler ordering of the nodes. Even though both implentations are independent of the ordering, the fattening method is definitely less complicated in general, and we imagine it would be preferred over the other method in practice, since they both have linear convergence order.

Part 2: A variable coefficient transport equation

For the transport problem we, in addition to initial conditions at $u(x,0)$, only need boundary conditions at the inflow boundary. In this case that is the left boundary, at $u(0,t)$. Thus, with a variable coefficient a, we end up with the equation

$$\begin{cases} u_t + a(x,t)u_x = 0 & \text{in } (0,1)\times(0,T), \\ u(x,0) = u_0(x) & \text{in } [0,1] \\ u(0,t) = g(t) & \text{in } [0,T] \end{cases}$$

The numerical Upwind scheme for internal points can be written as

$$U_m^{n+1} = (1 - r^+ - r^-)U_m^n + r^+ U_{m-1}^n + r^- U_{m+1}^n.$$

. Where $r^+ = max(r, 0)$, $r^- = (-r)^+$, $r = r^+ - r^-$ and $|r| = |a|\frac{k}{h}$.

We can simplify this to $\vec{U}^{n+1} = B\vec{U}^n$ with B = tridiag$(r^+, 1 - r^+ - r^-, r^-)$

Convergence

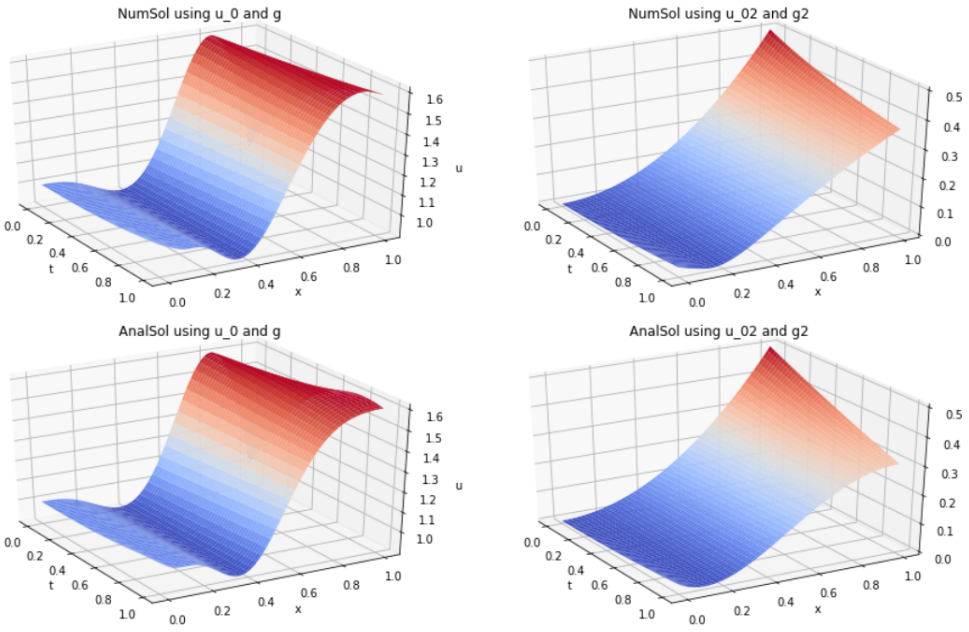


Figure 4: Numerical and analytic plots with $u_0(x) = 1 - sin(2\pi x)x^2$, $u_{02}(x) = \frac{x^2}{2}$, $g(t) = u_0(-a(0) \cdot t)$ and $g2(t) = u_{02}(-a(0) \cdot t)$

As for convergence plots we get

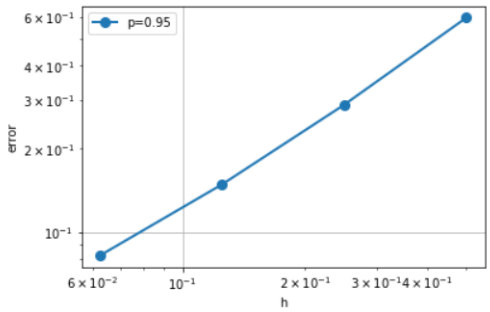


Figure 5: Spatial convergence for Upwind scheme

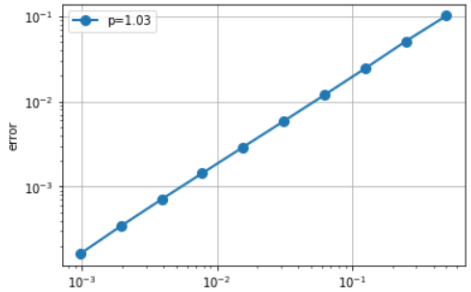


Figure 6: Temporal convergence for Upwind scheme

As we can see from the plots, the scheme converges both spatially and temporally in the first degree, this is in line with the theory.

Monotonicity and Von Neumann stability

For this scheme the monotonicity conditions becomes

- $1 \geq (1 - r^+ - r^-) + r^+ + r^-$
- $(1 - r^+ - r^-) \geq 0$
- $r^+ \geq 0$
- $r^- \geq 0$

From the second condition we get

$$\begin{aligned} r^+ + r^- &\leq 1 \\ |r| &\leq 1 \end{aligned}$$

Which then is our monotonicity condition.

To check Von Neumann stability we insert $U_m^n = \xi^n e^{i\beta x_m}$.

$$\begin{aligned} \implies \xi^{n+1} e^{i\beta x_m} &= (1-|r|)\xi^n e^{i\beta x_m} + r^+ \xi^n e^{i\beta(x_m - h)} + r^- \xi^n e^{i\beta(x_m+h)} \\ &= \xi^n e^{i\beta x_m}\big((1-|r|)r^+ e^{i\beta h} + r^- e^{-i\beta h}\big) \\ \implies \xi &= (1-|r|) + r^+ e^{i\beta h} + r^- e^{-i\beta h} \\ &= (1-|r|) + r^+\big(cos(\beta h) - isin(\beta h)\big) + r^-\big(cos(\beta h) + isin(\beta h)\big) \\ &= 1 - |r| + |r|cos(\beta h) - r \cdot isin(\beta h) \leq 1 \\ \implies |\xi|^2 &= Re^2 + Im^2 \\ &= \big(1 - |r| + |r|cos(\beta h)\big)^2 + \big(r \cdot sin(\beta h)\big)^2 \\ &= 1 - 2|r| + r^2 + 2|r|cos(\beta h) - 2r^2 cos(\beta h) + r^2 \leq 1 \\ &\implies r^2\big(1 - cos(\beta h)\big) - |r|\big(1 - cos(\beta h)\big) \leq 0 \\ &\implies r^2 - |r| \leq 0 \\ &\implies r \leq 1. \end{aligned}$$

Thus the Von Neumann stability condition is also $r \leq 1$.

Dissipativity and dispersion

Using $U_m^n = \rho^n e^{i(\omega t_n + \beta x_m)}$, the dissipative condition becomes $\rho < 1$. To find the specific condition we replace U in our scheme.

$$\begin{aligned} \rho^{n+1} e^{i(\omega(t_n+k)+\beta x_m)} &= (1-|r|)\rho^n e^{i(\omega t_n + \beta x_m)} + r^+ \rho^n e^{i(\omega t_n + \beta(x_m - h))} + r^- \rho^n e^{i(\omega t_n + \beta(x_m+h))} \\ &= \rho^n e^{i(\omega t_n + \beta x_m)}\big((1-|r|) + r^+ e^{-i\beta h} + r^- e^{i\beta h}\big) \\ \implies \rho e^{i\omega k} &= 1 + |r| + r^+\big(cos(\beta h) - i \cdot sin(\beta h)\big) + r^-\big(cos(\beta h) + i \cdot sin(\beta h)\big) \\ &= 1 + |r| + |r|cos(\beta h) - r \cdot i \cdot sin(\beta h) \end{aligned}$$

Now take absolute value and square both sides

$$\begin{aligned} \implies |\rho e^{i\omega k}|^2 &= Re^2 + Im^2 \\ |\rho|^2 &= \big(1 + |r| + |r|cos(\beta h)\big)^2 + \big(-r \cdot i \cdot sin(\beta h)\big)^2 \leq 1 \\ &= ... \\ \implies r &\leq 1. \end{aligned}$$

The last of the calculations follow exactly that done just above, to find Von Neumann stability condition. In other words, we find that the scheme is dissipative if $r \leq 1$.

Now to check if the scheme is dispersive, we look at whether the wavespeeds are constant or not. $v = \frac{\omega(\beta)}{\beta}$

$$\omega k = arg\big(Re + iIm\big) = arctan\big(\frac{Im}{Re}\big) = arctan\Big(\frac{-r \cdot sin(\beta h)}{1 - |r| + |r|cos(\beta h)}\Big)$$

From this we can see that ω does not constantly or linearly depend on β, thus, the scheme is dispersive.

Lax-Wendroff

The Lax-Wendroff scheme is a second order solution of the variable coefficient transport equation. To get the numerical scheme we first Taylor expand $u(x,t+k) = u + k \cdot u_t + \frac{1}{2}k^2 u_{tt} + O(k^3)$. We now replace the t derivatives using the formula for the transport equation:

$$\begin{aligned} u_t &= -a(x)u_x \\ u_{tt} &= -a\big(-a(x)u_x\big)_t \\ &= -au_{xt} - a_t u_x \\ &= -a(u_t)_x \\ &= -a\big(-au_x\big)_x \\ &= a^2 u_{xx} + aa_x u_x \end{aligned}$$

Thus we can rewrite and get the numerical scheme.

$$\begin{aligned} u(x,t+k) &= u + k \cdot u_t + \frac{1}{2}k^2 u_{tt} \\ &= u - ka(1 - \frac{1}{2}ka_x)u_x + \frac{1}{2}k^2 a^2 u_{xx} \\ \implies U_m^{n+1} &= U_m^n - \frac{1}{2}r(1 - \frac{1}{2}ka_x)(U_{m+1}^n - U_{m-1}^n) + \frac{1}{2}r^2(U_{m+1}^n - 2U_m^n + U_{m-1}^n) \\ &= (1-r^2)U_m^n + \big(\frac{1}{2}r^2 - \frac{1}{2}r(1 - \frac{1}{2}ka_x)\big)U_{m+1}^n + \big(\frac{1}{2}r^2 + \frac{1}{2}r(1 - \frac{1}{2}ka_x)\big)U_{m-1}^n \end{aligned}$$

For this scheme we did not get the correct degrees of convergence using the initial and boundary conditions used earlier in task 3. Since the problem might be that the scheme handles such a simple problem too fast to see the rate of convergence, we instead use a solution with a right hand side $f(x,t)$. Following is the computations to show how we implemented the right hand side.

$$\begin{aligned} u_t = au_x &=: f \\ \implies u_{tt} &= (f + au_x)_t \\ &= f_t + a_t u_x + au_{xt} \\ &= f_t + a_t u_x + a(f + au_x)_x \\ &= f_t + a_t u_x + a(f_x + a_x u_x + au_{xx}) \end{aligned}$$

Thus we when computing also need the derivatives of f in both time and space. We also chose the boundary condition at x=1 to be $U_M^{n+1} = U_{M-1}^{n+1}$.

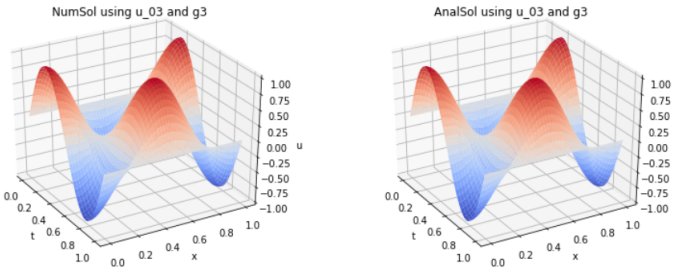


Figure 7: Lax-Wendroff numerical scheme and analytic plot using solution $u_0(x) = 1 - sin(2\pi x)x^2$ and $g(t) = u_0(-a(0) \cdot t)$

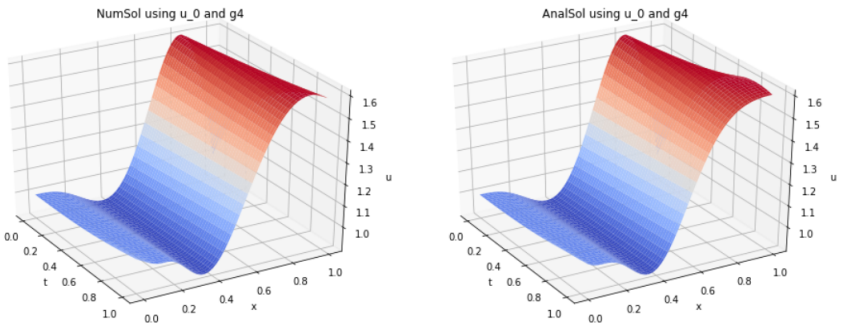


Figure 8: Lax-Wendroff numerical scheme and analytic plot using solution $u(x,t) = cos(2\pi x) \cdot np.sin(2\pi t)$

Convergence for Lax-Wendroff

The following convergence plots were found using a solution $u(x,t) = cos(2\pi x) \cdot np.sin(2\pi t)$ with initial condition $u(x,0) = 0$, boundary condition $g(t) = np.sin(2\pi t)$ and a corresponding right hand side $f = 2\pi cos(2\pi x) \cdot cos(2\pi t) - a(x) \cdot 2\pi sin(2\pi x) \cdot sin(2\pi t)$.

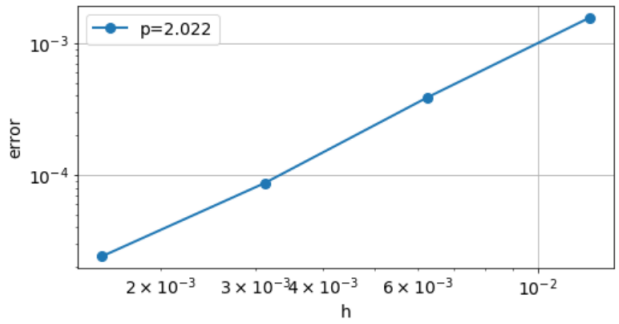


Figure 9: Spatial convergence for Lax-Wendroff scheme

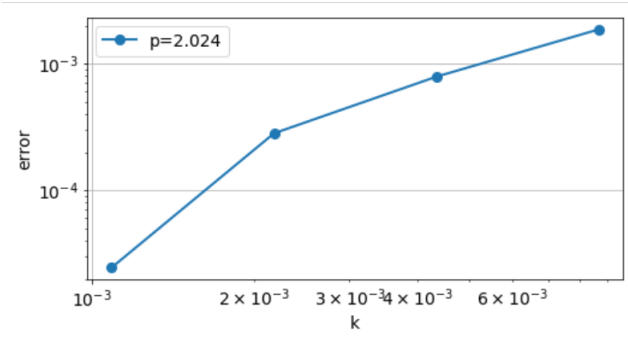


Figure 10: Temporal convergence for Lax-Wendroff scheme

These convergence plots are in accordance with theory, as we have from lectures that the scheme has an error of $O(h^2 + k^2 + kh^2)$.

Conclusion

In task two we found that both modified discretization and the method fatten the boundary solves the problem concerning irregular boundary. However fattening the boundary is both far easier to explain and implement numerically. From task three we found that Lax-Wendroff solves the same problem, the transport equation, with a higher order of accuracy compared to the Upwind scheme.