Project 1

M. Sandnes[a], A. Skogøy[a], P. Vårheim[a]

[a]*Dempartment of Mathematical Sciences, TMA 4212 - Numerical solution of differential equations by difference methods, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.*

1. Introduction

In this project we will do numerical analysis of the heat distribution in anisotropic- and isotropic materials, for both regular- and irregular grids. We explore the effect of irrational step sizes, along with none exact boundaries. We then implement different solvers for irregular grids, and compare the results.

2. Heat distribution in anisotropic materials

We start by experimenting with anisotropic materials on regular grids. We assume that the temperature inside the solid domain Ω, is represented by u and follows a Poisson distribution. The system is stationary, implying that $\partial_t u = 0$, with an internal heat source, f, and heat conductivity κ. Using Fourier's law for heat flux and that $\partial_t u = 0$ we obtain the following model for the problem

$$-\nabla \cdot (\kappa \nabla u) = f \quad \text{in } \Omega \tag{1}$$

For anisotropic materials the heat flow is faster in some directions than others. This is represented through κ as a matrix. We focus on a two dimensional model with a two dimensional heat-flow given by

$$\vec{d_1} = (1, 0) \quad \text{and} \quad \vec{d_2} = (1, r) \quad \text{where } r \in \mathbb{R}.$$

This gives us the following heat conductivity after normalization of the vectors

$$\kappa = \begin{pmatrix} a+1 & r \\ r & r^2 \end{pmatrix},$$

where $a > 0$ is a constant. Which gives the following simplification of the gradient of the heat-flux:

$$\nabla \cdot (\kappa \nabla u) = (a+1)\partial_x^2 u + 2r\partial_x \partial_y u + r^2 \partial_y^2 u = a\partial_x^2 u + (\vec{d_2} \cdot \nabla)^2 u, \tag{2}$$

thus equation (1) can be simplified to

$$-\left(a\partial_x^2 u + (\vec{d_2} \cdot \nabla)^2 u\right) = f \quad \text{in} \quad \Omega. \tag{3}$$

The reason why we use the directional derivative of second order, $(\vec{d_2} \cdot \nabla)^2$, instead of mixed derivatives is because discretization of mixed derivatives are numerically unstable.

We look at the case where $r = 1$, and we can discretize the derivatives using central differences in the following way

$$\partial_x^2 \approx \frac{1}{h^2}\delta_x^2, \quad \left(\vec{d_2} \cdot \nabla\right)^2 \approx \frac{1}{h^2}\delta_{\vec{d}_2}^2$$

where $\delta_x^2 u = u_{m+1,n} - 2u_{m,n} + u_{m-1,n}$ and $\delta_{\vec{d}_2}^2 u = u_{m+1,n+1} - 2u_{m,n} + u_{m-1,n-1}$. Thus we have that the scheme is given by

$$-\mathcal{L}_h U_{m,n} = -a\frac{U_{m+1,n} - 2U_{m,n} + U_{m-1,n}}{h^2} - \frac{U_{m+1,n+1} - 2U_{m,n} + U_{m-1,n-1}}{h^2} = f_{m,n} \tag{4}$$

To solve this equation we want to rewrite the scheme in matrix form, such that we can solve a linear system. To do this we choose the natural ordering of the system, such that for $(M+1) \times (N+1)$ grid points we get that

$$U_0 = U_{0,0}, U_1 = U_{1,0}, \ldots, U_M = U_{0,1}, U_{M+1} = U_{1,1}, \ldots, U_{M\cdot N} = U_{M,N},$$

and we transform the matrix $(U_{m,n})_{\substack{m=0,1,\ldots,M \\ n=0,1,\ldots,N}}$ into a $(M+1)\cdot(N+1)$ 1D array. We use Dirichlet boundary conditions for the system, i.e. $u_p = g_p$ for all points p on the boundary. Thus we can represent the system in matrix form as

$$A_h \vec{U} = \vec{f} + \frac{1}{h^2}\vec{g}, \tag{5}$$

where our $(M-1)(N-1) \times (M-1)(N-1)$ block triangular matrix A_h is given by

$$A_h = \frac{1}{h^2}\text{tridiag}\{-I_{M-1}, B, -I_{M-1}\},$$

where $B = \text{tridiag}\{-a, 2a+2, -a\}$, I_{M-1} is the $(M-1)\times(M-1)$ identity matrix and stepsize $h = \frac{1}{M}$. For big systems, A_h will be very sparse in the sense that most of the matrix elements are zero.

Using equation (4) we can draw a 5-point stencil for the problem using the notation of West (W), South-West (SW), East (E) and North-East (NE) points from the evaluated point (P). Thus we simplify the notation of the scheme to

$$-\mathcal{L}_h u_P = -a\frac{u_E - 2u_P + u_W}{h^2} - \frac{u_{NE} - 2u_P + u_{SW}}{h^2} = \alpha_{PP}u_P - \sum_{Q\neq P}\alpha_{PQ}u_Q = f_P,$$

where Q denotes one of the 4 different directions in the stencil. Later we want to show that this scheme is stable in L^∞, and thus we have to check if this scheme has positive coefficients:

$$\begin{aligned} &\alpha_{PQ} \geq 0 \\ &\alpha_{PP} \geq 0 \text{ and } \alpha_{PP} \geq \sum_{Q\neq P}\alpha_{PQ}. \end{aligned} \tag{6}$$

From the scheme we get that

$$\begin{aligned} &\alpha_{PQ} \geq 0, && \text{for all Q} \\ &\alpha_{PP} = (2a+2)/h^2 \geq 0 && (a>0) \\ &\alpha_{PP} = (2a+2)/h^2 \geq \sum_{Q\neq P}\alpha_{PQ} = (a+a+1+1)/h^2 = (2a+2)/h^2, \end{aligned}$$

thus we get that the scheme has positive coefficients. Since the scheme has positive coefficients we can use the discrete max principle (DMP) given by

$$-\mathcal{L}U_P \leq 0,\ P \in \overline{\mathbb{G}} \Rightarrow \max_{P\in\overline{\mathbb{G}}} U_P = \max_{P\in\partial\mathbb{G}} U_P, \tag{7}$$

where $\overline{\mathbb{G}}$ denotes the closure and $\partial\mathbb{G}$ denotes the boundary of the grid.

Using the properties of the DMP and a supersolution $\varphi \geq 0$ we can show that the scheme is stable in L^∞. We can choose a supersolution and V_p such that

$$\begin{cases} -\mathcal{L}_h\varphi \geq 1, & \text{in } \mathbb{G} \\ V_p := U_p - ||\vec{f}||_\infty \varphi_p, \end{cases}$$

then for Dirichlet boundary conditions ($U_p = g_p$, for all $p \in \partial\mathbb{G}$) we get that

$$\begin{cases} -\mathcal{L}_h V_p &= -\mathcal{L}_h U_p - ||\vec{f}||_\infty(-\mathcal{L}_h\varphi_p) \leq f_p - ||\vec{f}||_\infty \leq 0, \text{ for } p \in \mathbb{G} \\ V_p &:= U_p - ||\vec{f}||_\infty\varphi_p \leq g_p \leq |g_p|, \end{cases}$$

where we have used DMP for V_p to get the second result. Observe that

$$U_p = V_p + ||\vec{f}||_\infty\varphi_p \leq \max|g_p| + ||\vec{f}||_\infty \max\varphi_p,$$

and similarly

$$-U_p \leq \max|-g_p| + ||-\vec{f}||_\infty \max \varphi_p,$$

thus we get that

$$\Rightarrow \max_{\overline{\mathbb{G}}} |U_p| \leq \max_{\partial\mathbb{G}} |g_p| + \max_{\partial\mathbb{G}} \varphi_p ||\vec{f}||_\infty. \tag{8}$$

Now we need to find a supersolution φ such that

$$\begin{cases} -\mathcal{L}_h\varphi \geq 1 & \text{in } [0,1]^2 \\ \varphi \geq 0 & \text{in } [0,1]^2, \end{cases}$$

thus we can try $\varphi(x) = \frac{1}{2}x(1-x)$, which clearly satisfies $\varphi(x) \geq 0$ on the grid. Note that $\partial_y\varphi = 0$ and the error in the derivative approximation $e = (\partial_x^2 - \frac{1}{h^2}\delta_x^2)\varphi(x) = \partial_x^4\varphi(\xi) = 0$. Thus we get that

$$-\mathcal{L}_h\varphi = -a\frac{1}{h^2}\delta_x^2\varphi - \frac{1}{h^2}(\delta_{\vec{d}_2})^2\varphi = -a\partial_x^2\varphi - \partial_x^2\varphi = a + 1 \overset{(a>0)}{\geq} 1,$$

where $(\delta_{\vec{d}_2})^2$ denotes the second order approximation to the derivative in the direction $\vec{d}_2 = (1,1)$. The result above verifies that the supersolution is OK! Then by finding the max of the function $\varphi(x)$ and using equation (8) we obtain that

$$||\vec{U}||_\infty \leq \max_{\partial\mathbb{G}} |g_p| + \frac{1}{8}||\vec{f}||_\infty \quad \text{where } U_p = g_p \text{ on } \partial\mathbb{G}, \tag{S}$$

and we can conclude that the scheme is stable in L^∞. We can use (S) to find an upper bound for the error $e_p = u_p - U_p$, by using the scheme on e_p:

$$-\mathcal{L}_h e_p = -\mathcal{L}_h u_p - (-\mathcal{L}_h U_p) = (-\mathcal{L}u_p - \tau_p) - f_p = f_p - \tau_p - f_p = -\tau_p$$

$$((\text{S}) \text{ and } e_p = 0, p \in \partial\mathbb{G}) \Rightarrow ||\vec{e}||_\infty \leq \frac{1}{8}||\vec{\tau}||_\infty$$

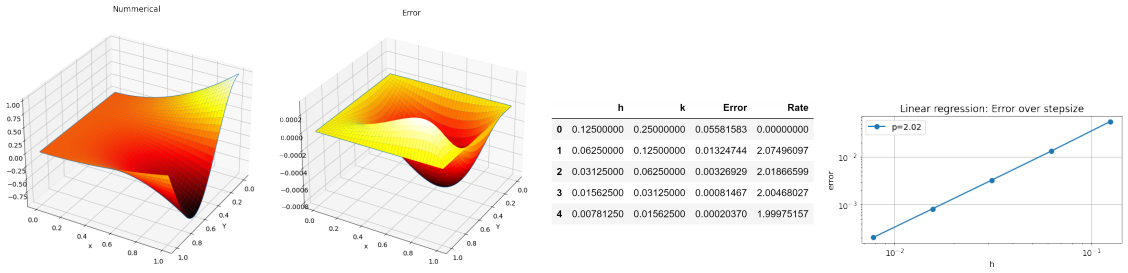
where the local truncation error $\tau = -\mathcal{L}u_p - (-\mathcal{L}_h u_p)$ can be found by taylor expansion of the scheme:

$$\begin{aligned}
-\mathcal{L}_h u_p &= -\frac{a}{h^2}\delta_x^2 u_p - \frac{1}{h^2}\delta_{\vec{d}_2}^2 u_p \\
&= -\frac{a}{h^2}\left(2\cdot\frac{h^2}{2!}\partial_x^2 u_p + 2\cdot\frac{h^4}{4!}\partial_x^4 u(\xi,y)\right) \\
&\quad - \frac{1}{h^2}\left(2\cdot\frac{h^2}{2!}\left(\partial_x^2 u_p + 2\partial_x\partial_y u_p + \partial_y^2 u_p\right) + 2\cdot\left(\frac{h^4}{4!}\partial_x^4 u_p + \frac{h^4}{3!1!}\partial_x^3\partial_y u_p + \frac{h^4}{2!2!}\partial_x^2\partial_y^2 u_p + \frac{h^4}{1!3!}\partial_x\partial_y^3 u_p + \frac{h^4}{4!}\partial_x^4 u_p\right)\right) \\
&= -a\partial_x^2 u_p - \frac{ah^2}{12}\partial_x^4 u(\xi,y) \\
&\quad - \left(\partial_x^2 u_p + 2\partial_x\partial_y u_p + \partial_y^2 u_p\right) - h^2\left(\frac{1}{12}\partial_x^4 u(\xi,y) + \frac{1}{3}\partial_x^3\partial_y u(\xi,\eta) + \frac{1}{2}\partial_x^2\partial_y^2 u(\xi,\eta) + \frac{1}{3}\partial_x\partial_y^3 u(\xi,\eta) + \frac{1}{12}\partial_y^4 u(x,\eta)\right) \\
&= -\left((a+1)\partial_x^2 u_p + 2\partial_x\partial_y u_p + \partial_y^2 u_p\right) \\
&\quad - h^2\left(\frac{a+1}{12}\partial_x^4 u(\xi,y) + \frac{1}{3}\partial_x^3\partial_y u(\xi,\eta) + \frac{1}{2}\partial_x^2\partial_y^2 u(\xi,\eta) + \frac{1}{3}\partial_x\partial_y^3 u(\xi,\eta) + \frac{1}{12}\partial_y^4 u(x,\eta)\right) \\
&= -\mathcal{L}u_p - Ch^2 \\
\Rightarrow \tau_p &= -\mathcal{L}u_p - (-\mathcal{L}_h u_p) = Ch^2, \text{ where } C \text{ is given by} \\
C &= \left(\frac{a+1}{12}\partial_x^4 u(\xi,y) + \frac{1}{3}\partial_x^3\partial_y u(\xi,\eta) + \frac{1}{2}\partial_x^2\partial_y^2 u(\xi,\eta) + \frac{1}{3}\partial_x\partial_y^3 u(\xi,\eta) + \frac{1}{12}\partial_y^4 u(x,\eta)\right)
\end{aligned}$$

We observe that the truncation error τ is of second order $o(h^2)$, and thus the scheme has convergence rate of $p = 2$. If we let $K = ||C||_{L^\infty}$, then we can finally get an error bound for the scheme

$$||\vec{e}||_\infty \leq \frac{1}{8}||\vec{\tau}||_\infty = \frac{1}{8}Kh^2 \tag{9}$$

To test that the scheme has convergence of second order, we have to implement it numerically. The plots for the scheme are shown below in figure (1).



	h	k	Error	Rate
0	0.12500000	0.25000000	0.05581583	0.00000000
1	0.06250000	0.12500000	0.01324744	2.07496097
2	0.03125000	0.06250000	0.00326929	2.01866599
3	0.01562500	0.03125000	0.00081467	2.00468027
4	0.00781250	0.01562500	0.00020370	1.99975157

Figure 1: The plots to the left shows the numerical solution and error for the function $u(x, y) := x^2 \cos(5y)$, with M=N=40, $a = 2$. The plots to the right shows the numerical error $e = u - U$ and the convergence rate p.

The figure shows that the numerical error for the system is of order 10^{-4}, which is acceptable. We observe that the numerical convergence rate is of order $p = 2$, as deduced analytically.

Now we introduce a new grid with step sizes $h = \frac{1}{M}$ and $k = |r|h$ in the x and y directions, and let r be an irrational number. By combining these definitions, we calculate the number of grid points in the y-direction.

$$N = \frac{1}{k} = \frac{1}{|r|h} = \frac{M}{|r|} \Leftrightarrow |r| = \frac{M}{N}, \quad M, N \in \mathbb{N}$$

Since r is an irrational number it can never be expressed as a fraction. Therefore we must approximate N to the upper integer value of the fraction $\frac{M}{|r|}$. This ensures that $Y_{max} = k\dot{N} > 1$, which means the upper bound of the grid, $U_{x,N}$, will be outside the domain $\Omega = [0, 1]^2$. Since the problem is undefined outside Ω, the last grid level has to be "moved down" from $Y = kN$ to $Y = 1$, and thereby adjusting the distance between $U_{x,N-1}$ and $U_{x,N}$ from k to $k' = 1 - (k(N - 1))$. To achieve this we implement a smooth extension of the boundary, meaning we extend the boundary to include the necessary nodes outside Ω. We now have to assign a value to each of these grid points. This can be solved by either assuming the exact values of $U_{x,N}$ are known, or by using a projection for the values at $U_{x,N}$.

When solving with the exact values at all boundary points, we are practically solving the same problem as for the none irrational r. Since the values at $U_{x,N}$ will be used when solving the linear system of equation, the distance between all levels of grid-points will be equal to k. We therefore expect a convergence-rate of 2, which is confirmed by the numerical results 2.

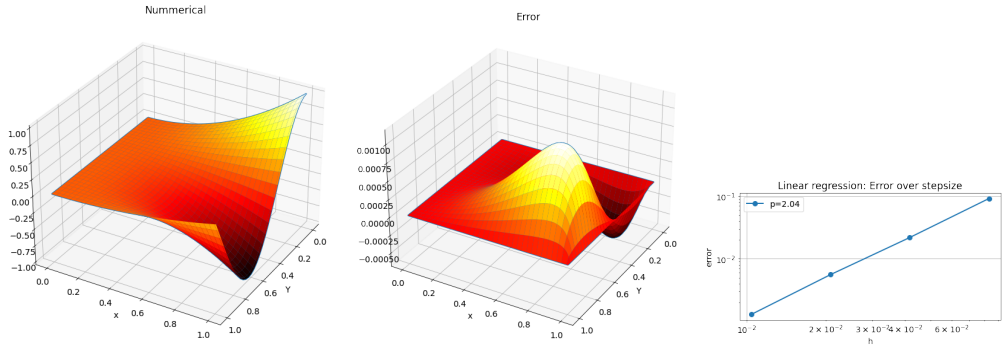


Figure 2: The plot to the left shows the numerical solution of the function $u(x, y) := x^2 \cos(5y)$, with M=100, $r = \pi$, a=2. The plot in the middle shows the numerical error $e = u - U$ and the right image shows the convergence rate $p = 2.04$.

It is not always the case that the grid values outside the domain is known. To account for this we use a projection to approximate the values in these nodes. We use the closest point projection by projecting the values at $y = 1$ to the values at $y = kN$. Geometrically, we adjust the distance between $U_{x,N-1}$ and $U_{x,N}$ to be k'. This will effect our solution as we assume the distance between the nodes to be constantly equal to k.

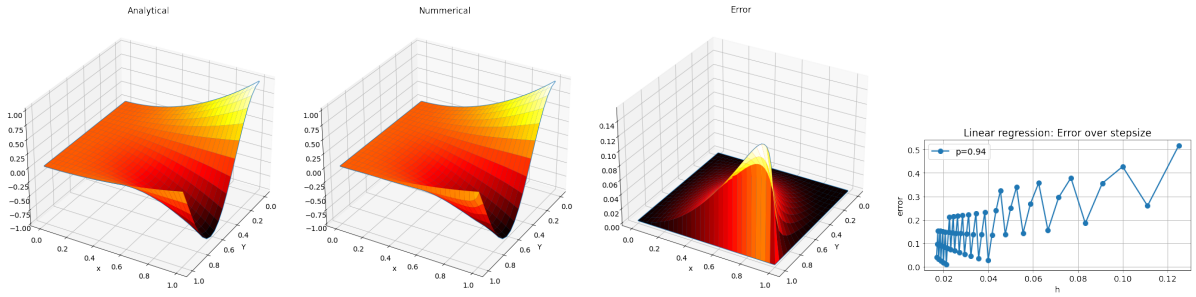


Figure 3: The plots to the left shows the analytical and numerical solution of the function $u(x, y) := x^2 \cos(5y)$, with M=60, $r = \pi$, a=2. The plots to the right shows the numerical error $e = u - U$ and how the max error changes when increasing M by 1. We increase from $M = 8$ to $M = 60$.

As we see from figure (3) the largest error will occur at $U_{x,N-1}$. The largest error is located in the north-east corner, which makes sense since this is the node most effected by the projection. In the linear regression plot we observe a peculiar pattern. This however, can be explained by the "projection distance". As we test for several different values of M the distance between $y = kN$ and $y = 1$ varies. In some cases we are "lucky" cause we get a smaller distance then we do in other cases. This explains why we observe a pattern with spikes. The top of the spike represents an unlucky value of h, and the bottoms represents a lucky value of h. As M increases the distance between the lucky and unlucky values decreases, which makes sense since the maximum projection-distance decreases with h. Even though the error doesn't decrease completely linearly, we observe a steady decrease of error. In figure (3), we get a decrease of order $p = 0.94$. Since the impact of projection distance decreases with the value of M, we would expect the linearity of the function to increase with M. As M approaches a larger value, we would therefore expect a convergence rate closer to 1. The difference in convergence rate makes fattening the boundary with the exact values at $U_{x,N}$ the preferable method. This however, demands knowledge of the boundary function outside the original domain. This makes the projection solution more applicable.

3. Irregular boundaries

We look at the system with domain Ω which is enclosed by the three curves

$$\gamma_1 = [0, 1] \times \{0\}, \quad \gamma_2 = \{0\} \times [0, 1], \quad \text{and} \quad \gamma_3 = (x, h(x)) : x \in [0, 1],$$

where $h(x) = \frac{1}{2}(\cos(\pi x) + 1)$. To simplify the problem we now assume that the material is isotropic. Then the heat conductivity $\kappa = I$ and the equation simplifies to

$$-\nabla \cdot (\kappa \nabla u) = -\Delta u = f,$$

where $\Delta = \partial_x^2 + \partial_y^2$. As before we use central difference, and get the following discretization

$$-\mathcal{L}_h U_P = -\left(\frac{U_E - 2U_P + U_W}{h^2} + \frac{U_N - 2U_P + U_S}{h^2}\right) = -\frac{1}{h^2}\left(U_E + U_W + U_N + U_S - 4U_P\right) = f_P, \quad P \in \Omega.$$

It follows as before that this satisfies the DMP, if we assume that U_{P^*} is a maximum of U on the grid

$$\begin{aligned} -\mathcal{L}_h U_{P^*} &\leq 0 \\ \Rightarrow -\mathcal{L}_h U_{P^*} &= \frac{1}{h^2} \sum_{Q=N,E,S,W} (U_{P^*} - U_Q) \leq 0 \end{aligned}$$

which means that $U_Q = U_{P^*}$ for each direction $Q = N, W, E, S$. Choose the new U_{P*} to be one of the U_Q's, continue in this manner until we hit the boundary, and we can conclude that

$$\max_{P \in \overline{\mathbb{G}}} U_P = \max_{P \in \partial \mathbb{G}} U_P, \quad \text{when } -\mathcal{L}_h U_P \leq 0.$$

Using similar calculation as for the anisotropic case we also get that this scheme is stable in L^∞, but we will not prove this here.

For the right hand side boundary $h(x)$ we have to approximate the derivatives such that the truncation error τ is of at least order $o(h)$. We can do this by

1) Changing the discretization near the boundary
2) Fattening the boundary

We look at 1) first:
For the discretization in the x-direction we have to solve the following equation such that $\tau_{x,P} = o(h)$:

$$a_W u_W + a_P u_P + a'_E u'_E = \partial_x u_P - \tau_{x,P} \tag{10}$$

where u'_E denotes the boundary point to the east. We rewrite u'_E such that $u'_E = u(x + h\xi, y)$ for some $\xi \in (0, 1)$. Then we can Taylor expand (10) and solve it such that $\tau_x = o(h)$:

$$a_W \left(u - hu_x + \frac{1}{2}h^2 u_{xx} + o(h^3) \right) + a_{p,x} u + a'_E \left(u + \xi h u_x + \frac{1}{2}(\xi h)^2 u_{xx} + o(h^3) \right) = \partial_x^2 u - \tau_x$$

$$\Rightarrow \begin{cases} a_W + a_{p,x} + a'_E = 0 \\ -h a_W + \xi h a'_E = 0 \\ \frac{1}{2}h^2 a_W + \frac{1}{2}(\xi h)^2 a'_E = 1 \end{cases} \Leftrightarrow \begin{cases} a_{p,x} = -\frac{2}{h^2 \xi} \\ a_W = \frac{2}{h^2(1+\xi)} \\ a'_E = \frac{2}{h^2 \xi(1+\xi)} \end{cases},$$

which also gives the truncation error $\tau_x = o(h^{-2}) \cdot o(h^3) = o(h)$ as we wanted. Similarly for the y-direction, using that $u'_N = u(x, y + \eta k)$, $\eta \in (0, 1)$, we get that

$$\begin{cases} a'_N + a_{p,y} + a_S = 0 \\ -k a_S + \eta k a'_N = 0 \\ \frac{1}{2}k^2 a_S + \frac{1}{2}(\eta k)^2 a'_N = 1 \end{cases} \Leftrightarrow \begin{cases} a_{p,y} = -\frac{2}{k^2 \eta} \\ a_S = \frac{2}{k^2(1+\eta)} \\ a'_N = \frac{2}{k^2 \eta(1+\eta)} \end{cases}.$$

Thus if we have a point $P \in \Omega$ where the the north and east point both are outside Ω we can use the following discretization

$$-\mathcal{L}_h U_P = -\left(\frac{a'_E U'_E + a_{p,x} U_P + a_W U_W}{h^2} + \frac{a'_N U'_N + a_{p,y} U_P + a_S U_S}{k^2} \right),$$

and if only the east point is outside Ω we discretize the y-direction as usual, and vice versa. The truncation error τ for the discretization will at worst be $\tau = \tau_x + \tau_y = o(h) + o(k) = o(h + k)$.

For method 2) we have to make a smooth extension of the function g at the boundary. An easy way to overcome this problem is to choose a function g which is defined in all of $\mathbb{R}^2$. Then we can use g as its own extension. Thus if we have a point P near the right hand side boundary, where the north and east point are outside Ω, we can use that $U_N = g(N)$ and $U_E = g(E)$, and the following discretization can be used

$$-\mathcal{L}_h U_P = -\left(\frac{g(E) - 2U_P + U_W}{h^2} + \frac{g(N) - 2U_P + U_S}{k^2} \right).$$

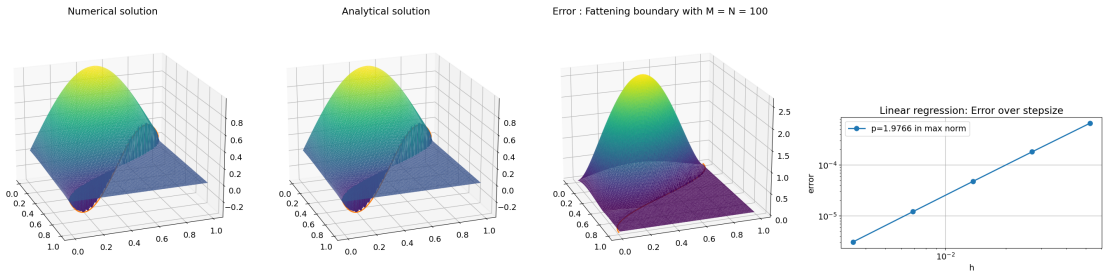


Figure 4: The figure shows the numerical solution, analytical solution, $u(x, y) = \cos(\pi x)\sin(\pi y)$, error $e = u - U$ and the convergence rate $p = 1.9766$ for fattening boundary with $M = N = 100$.

Observe that we get a convergence rate $p = 1.9766$, which supports the analytical result $p = 2$. It's worth noting that the method requires that the function, g, is defined close to the boundary. This is not always

the case. For example if we want to compute the temperature of a gas in a heat isolated container. We are no longer able to extend g outside the container. An alternative solution here is to use fattening boundary with projection. This will give a truncation error of $o(h)$ for every smooth extension of g.

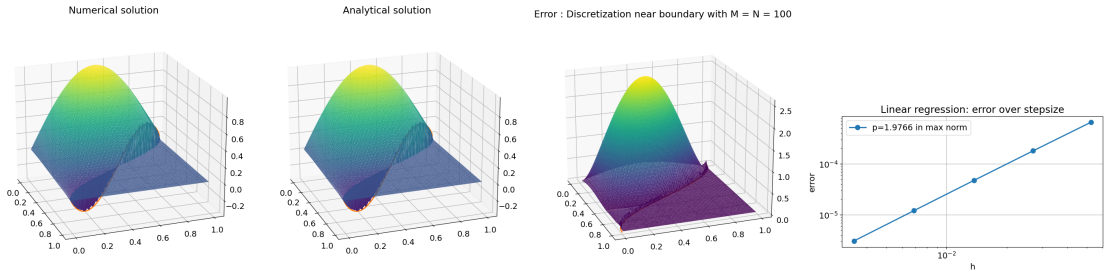


Figure 5: The figure shows the numerical and analytical solution of $u(x, y) = \cos(\pi x)\sin(\pi y)$, error $e = u - U$ and the convergence rate $p = 1.9766$ for changed discretization near boundary with $M = N = 100$

In figure (5) we use changed discretization near the boundary and expect to get convergence rate $p = 1$ for error along the boundary $h(x)$. We observe a small spike in error along the $h(x)$ boundary, while the dominant error occurs in the interior of Ω. From figure (5) we observe that the convergence is of order $p = 2$, which seems to be the case for most of the test functions. While this initially seems odd since the derived convergence rate is of order $p = 1$, we remember that this only holds for the boundary nodes. The interior nodes will still have a convergence rate of $p = 2$, and since the maximum error lies in the interior of the domain, a convergence rate of second order makes sense. The interior error decreases by h^2 and the boundary error decreases by h. At some point the boundary error will surpass the interior error, and thereby changing the rate of convergence to $p = 1$. Theoretically this is possible if we choose M large enough, but we dont have the computing power necessary to achieve this.

When comparing the two methods we observe some advantages and disadvantages with each of the methods. The fattening boundary method is substantially easier to implement, since there is no regulation of the step size. We also avoid the error spike along the boundary-function $h(x)$. The main weakness of the method is its dependence on a boundary function g. If g is only defined directly on the boundary, finding a smooth extension of g is not always possible. Changing the discretization near boundary method is more complicated to implement, because we have to compute a specialized stencil each time we breach the boundary. The advantage of this method is that our stencil never breaches the domain. This makes the method more applicable. We see no substantial difference in computational complexity between the methods.

4. Summary

In this project we have implemented different finite difference methods to find numerical solutions to the heat distribution. For regular grid with rational step-sizes we get a convergence-rate of second order. When solving problems with irrational step sizes by extension of the boundary, we get a convergence rate tending to 1 by increasing grid sizes. For irregular grids we found it easier to implement fattening of the boundary, then changing the democratization near the boundary. For both methods we get a convergence rate of order $p = 2$ and see no notable difference in complexity.

5. Appendix

Figure (.6) shows the domain used in the section regarding the irregular domain.

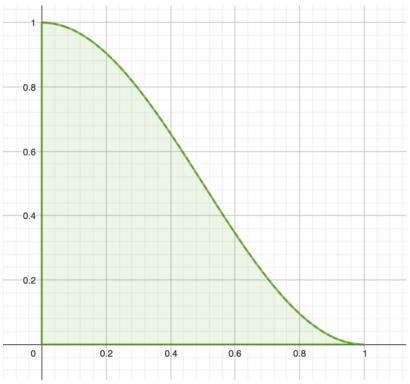


Figure .6: The figure shows the domain Ω