Project 1 TMA4212 - Solving the heat equation

Reidar Bråthen Kristoffersen, Viktor Sandve, Trond Skaret Johansen

September 30, 2024

Abstract

In this paper we present or findings from numerically solving the anisotropic heat equation in two dimensions, using finite difference methods. We experiment with how our numerical solution reacts to changes in the model. We also compare two approaches for handling a non-rectangular boundary; namely fattening the boundary and modification of discretisation near the boundary.

1 Introduction

The heat equation is the main tool for modelling the stationary distribution of the temperature, T, in a domain Ω. However in most real systems heat flow is not the same in all directions. In our model we assume that there are two directions of heat flow in the material, $\vec{d_1} = (1, 0)$ and $\vec{d_2} = (1, r)$, $r \in \mathbb{R}$. We then assume to know the heat conductivity matrix $\kappa = \begin{pmatrix} a+1 & r \\ r & r^2 \end{pmatrix}$ for some $a > 0$ We can now analyse the anisotropic heat equation given by

$$-\nabla \cdot (\kappa \nabla T) = -a\partial_{xx}T - (\vec{d} \cdot \nabla)^2 T = f \,\text{in}\, \Omega. \tag{1}$$

Where f is the energy density. Using different f, along with some boundary conditions, we can solve the equation both analytically and numerically over the domain to compare the accuracy we achieve under different assumptions for the models.

2 Theory and methods

2.1 Model for anisotropic materials on a rectangular grid

2.1.1 Discretisation

We organise the grid, $\bar{\mathbb{G}} = \mathbb{G} \cup \partial\mathbb{G}$, with step sizes $h = \frac{1}{M}$ in the x-direction and hr in the y-direction, and use notation $u_m^n := u(mh, nhr)$ or $u_P = u(x, y), (x, y) \in \bar{\mathbb{G}}$. Further, we let U_m^n, denote the numerical solution approximating u_m^n. Using a row-wise ordering we structure function evaluations of internal points in a vector $\vec{u} = (u_1^1, u_2^1, ..., u_1^2, u_2^2, ...)$.

To dicretise Equation (1) we need a discretisation of the directional derivative $(\vec{d_2} \cdot \nabla)^2 T$. Using the method of undetermined coefficients we seek the solution with maximal precision. For our central difference we use the values U_{m-1}^{n-1}, U_m^n and U_{m+1}^{n+1}. We need to find A, B, C such that $u_m^n = Au_{m-1}^{n-1} + Bu_m^n + Cu_{m+1}^{n+1} + \mathcal{O}(h^r)$ for maximal r. Writing out the truncation error, using

Taylor expanding u_{m-1}^{n-1} and u_{m+1}^{n+1}, denoting $u_m^n = u$ and reorganising we find:

$$\begin{aligned}\tau =&Au_{m-1}^{n-1} + Bu_m^n + Cu_{m+1}^{n+1} - (\vec{d_2}\cdot\nabla)^2 u\\ =&(A+B+C)u + h(A-C)(u_x + ru_y) + h^2\left(\frac{A}{2}+\frac{C}{2}-\frac{1}{h^2}\right)(u_{xx}+2ru_{xy}+r^2u_{yy}) + \mathcal{O}(h^3).\end{aligned}$$

Choosing $A = 1/h^2, B = -2/h^2, C = 1/h^2$ is the unique solution such that the lower order terms vanish. Similarly, one can deduce the usual central difference discretisation:

$$\partial_{xx}u_m^n \approx \frac{U_{m+1}^n - 2U_m^n + U_{m-1}^n}{h^2}.$$

Putting it all together, we obtain the scheme:

$$\nabla\cdot(\kappa\nabla u_m^n) \approx \mathcal{L}_h U_m^n = \frac{U_{m+1}^{n+1} + U_{m-1}^{n-1} + aU_{m+1}^n + aU_{m-1}^n - (2a+2)U_m^n}{h^2}. \tag{2}$$

The stencil can be found in Figure 5a.

2.1.2 Error analysis

Definition 1. A scheme of the form

$$-\mathcal{L}_h u_P = \alpha_{PP}u_P - \sum_{Q\neq P}\alpha_Q u_Q, \quad P\in\mathbb{G}, \quad Q\in\bar{\mathbb{G}}$$

is said to have *positive coefficients* if $\alpha_{PP} > 0, \alpha_{PQ} \geq 0, \alpha_{PP} \geq \sum_{Q\neq P}\alpha_Q, \quad P\in\mathbb{G}, \quad Q\in\bar{\mathbb{G}}$.

Proposition 1. *The scheme in Equation* (2) *has positive coefficients.*

Proof. From Equation (2) we obtain the coefficients:

$$\alpha_{PP} = \frac{1}{h^2}(2a+2), \quad \alpha_{PQ_1} = \alpha_{PQ_2} = \frac{1}{h^2}a, \quad \alpha_{PQ_3} = \alpha_{PQ_4} = \frac{1}{h^2}.$$

Indeed, all coefficients are greater than 0, and $\sum_{Q\neq P}\alpha_Q = \frac{1}{h^2}(2a+2) = \alpha_{PP}$. I. e. scheme 2 has positive coefficients. □

Definition 2. A scheme is said to be *boundary connected* if for each $P\in\mathbb{G}$ there is $Q\in\partial^*\mathbb{G}\subset\partial\mathbb{G}$ and $P_1,...,P_k\in\mathbb{G}$ such that $\alpha_{PP_1},\alpha_{P_1P_2},...,\alpha_{P_{k-1}P_k}\neq 0$, where $\alpha_{P_iP_j}$ are given as in Definition 1.

Proposition 2. *The scheme in Equation* (2) *is boundary connected.*

We omit a formal proof of this, but it can be seen by choosing any point in the grid and choosing any of the directions with non-zero coefficients in the scheme. Continually moving in the chosen direction the path will eventually reach a boundary point, due to the rectangular grid.

Our motivation for introducing the concepts from definitions 1 and 2 is the theorem below, which will be important in the error analysis of the scheme 2.

Theorem 1. (Discrete Maximum Principle) *For any boundry conneted scheme with positive coefficients, we have:*

$$-\mathcal{L}_h u_P \leq 0, \quad P\in\mathbb{G} \implies \max_{P\in\mathbb{G}} u_P \leq \max_{P\in\partial^*\mathbb{G}\subset\partial\mathbb{G}}\{0, u_P\}.$$

We will not include a proof of this theorem, but it can be found in [1, p. 76-77]. Now we introduce the concept of stability, before finally deriving an error bound and proving convergence of the scheme.

Definition 3. A scheme is said to be *stable* in L^∞, if for any solution U and right hand side f, given boundary conditions $U_P = 0 \quad \forall P \in \partial\mathbb{G}$,

$$\|\vec{U}\|_\infty \leq C\|\vec{f}\|_\infty,$$

for some constant C independent of the step size h.

Proposition 3. *The scheme in equation 2 is stable in* L^∞

Proof. Let U be a solution to the scheme with right hand side f, and 0 on the boundary. Now define the function $\phi_P = \frac{1}{2}x(1-x), P = (x, y)$, note that $\phi_P > 0$ when $x \in (0, 1)$ and $\max_{x\in[0,1]} \phi(x, y) = \frac{1}{8}$. Applying the scheme to ϕ_P we find

$$\begin{aligned} -\mathcal{L}_h\phi_P =& \frac{1}{h^2}(2a(\frac{1}{2}x(1-x)) - a\frac{1}{2}(x+h)(1-(x+h)) - a\frac{1}{2}(x-h)(1-(x-h)) + 2\frac{1}{2}x(1-x) \\ &- \frac{1}{2}(x+h)(1-(x+h)) - \frac{1}{2}(x-h)(1-(x-h))) = a+1 \geq 1. \end{aligned}$$

Where the last equality is from writing out parenthesis and simplifying. Now we consider the function $v_P = U_P - \|\vec{f}\|_\infty\phi_P$. Applying the scheme we get:

$$-\mathcal{L}_h v_P = -\mathcal{L}_h U_P - (-\|\vec{f}\|_\infty \mathcal{L}_h\phi_P) = f_P - \|\vec{f}\|_\infty(a+1) \leq 0.$$

Since the scheme is boundary connected and has positive coefficients we apply Theorem 1 to get:

$$\max_{P\in\mathbb{G}} v_P \leq \max_{P\in\partial^*\mathbb{G}\subset\partial\mathbb{G}}\{0, v_P\} = 0 \qquad \Longrightarrow \qquad \max_{P\in\mathbb{G}} U_P \leq \|\vec{f}\|_\infty \max_{P\in\mathbb{G}}\phi_P \leq \frac{1}{8}\|\vec{f}\|_\infty.$$

Now we note that $-U$ is a solution to the scheme with right hand side $-f$. Through the same procedure as above we get:

$$\max_{P\in\mathbb{G}} U_P \leq \frac{1}{8}\|\vec{-f}\|_\infty = \frac{1}{8}\|\vec{f}\|_\infty. \tag{3}$$

Since we have an upper bound on both the max of U_P and of $-U_P$, this implies $\|\vec{U}\|_\infty \leq 1/8\|\vec{f}\|_\infty$. □

Now lets consider the local truncation error, that is, the error obtained when applying the scheme to the exact solution.

$$\tau^n_m = \mathcal{L}_h u^n_m - \mathcal{L}u^n_m = \frac{1}{h^2}(u^{n+1}_{m+1} + u^{n-1}_{m-1} + a(u^n_{m+1} + u^n_{m-1}) - (2+2a)u^n_m) - a\partial^2_x u - (\vec{d_2}\cdot\nabla)^2 u.$$

Taylor expanding the first four terms and simplifying we get:

$$\tau^n_m = \frac{a}{12}h^2\partial^4_x u(\xi_{1,m}, y_n) + \frac{1}{12}h^2(\vec{d_2}\cdot\nabla)^4 u(\xi_{2,m}, \eta_n), \quad \xi_{1,m}, \xi_{2,m} \in [x_m, x_m + h], \eta_n \in [y, y+h].$$

Given sufficiently smooth solutions, this expression can be bound for all points in the grid, resulting in:

$$\|\vec{\tau}\|_\infty \leq \frac{1}{12}h^2(a\|\partial^4_x u\|_{L^\infty} + \|(\vec{d_2}\cdot\nabla)^4 u\|_{L^\infty}).$$

Finally we define the error $e_P = u_P - U_P$. Applying the scheme to e_P we find:

$$-\mathcal{L}_h e_P = -\mathcal{L}_h u_P - (-\mathcal{L}_h U_P) = -(\mathcal{L}_h u_P - \mathcal{L} u_P) - (-\mathcal{L}_h U_P) - \mathcal{L} u_P = -\tau_P - f_P + f_P = -\tau_P.$$

Hence e_P is a solution to the scheme with $-\tau_P$ as right hand side. It has value 0 along the boundary since $u_P = U_P = g_P$ here. Therefore we can utilize the stability of the scheme and the bound on the truncation error given in Equation (3). Assuming smooth solution, we then get the error bound:

$$\|\vec{e}\|_\infty \leq \frac{1}{8}\|\vec{\tau}\|_\infty \leq \frac{1}{96}h^2(a\|\partial_x^4 u\|_{L^\infty} + \|(\vec{d_2} \cdot \nabla)^4 u\|_{L^\infty}) \underset{h \to 0}{\longrightarrow} 0. \tag{4}$$

From the error bound we can see that, for smooth solutions, the scheme converges with rate 2.

2.1.3 Irrational r

Since we consider any $r \in \mathbb{R}$, we have in particular the case when r is irrational. Let $h = 1/M$ be the step size in the x-direction and $k = rh$ be the step size in the y-direction. Then there is no integer L such that $kL = rL/M = 1$, in other words we miss the upper boundary. This is easily handled when the boundary function is defined for $y > 1$. Otherwise, we take the value obtained by projection onto the boundary. A final option is to modify the discretisation, which we only explore for the irregular domain.

2.2 Non-rectangular boundary

For our final experiments we consider a non rectangular boundary. Our domain Ω is given by one of the enclosure of the coordinate axis and the graph of the function $h(x) = \frac{1}{2}(\cos(\pi x) + 1)$. The domain can be seen in Figure 1. When working with this domain, we consider the isotropic case, i.e. $\kappa = I$. Equation (1) then simplifies to $\nabla^2 T = f$. We explore two methods for handling the boundary.

2.2.1 Fattening the boundary

The idea of the method of fattening of the boundary is to take grid points outside the boundary. The closest ones lie within distance $R = \sqrt{h^2 + k^2}$ of $\partial\Omega$. The grid points and boundary points for this scheme is shown in Figure 1. We first make the assumption that the boundary condition g is defined outside the boundary. Then we simply evaluate g in the external boundary points. However this may not always be the case, and we need to obtain the value in the external point P in another manner. We then compute the projection $\pi(P)$ as the solution of

$$\pi(P) = \underset{Q \in \partial\Omega}{\arg\min} \|P - Q\|^2. \tag{5}$$

With this, we approximate $U_P \approx g(\pi(P))$, and note that this introduces an error of $\mathcal{O}(h)$.

2.2.2 Modification of the discretisation near the boundary

Another remedy for the irregular boundary is to move the external grid points onto the boundary and work with varying step sizes.

We turn to the task of discretising $\partial_{xx} u$ and $\partial_{yy} u$ with varying step size. Consider the case of $\partial_{xx} u_m^n$ at the point $P = (x_m, y_n)$. Denote $u_m^n = u$. For our problem, we always have constant left step size. The right one may shrink by a factor we denote s. Thus we have step sizes h, sh in the left and right direction respectively. By the method of undetermined coefficients, we need to find A, B, C such that $\tau = Au_{m-1}^n + Bu_m^n + Cu_{m+1}^n - u_m^n$ is of maximal order. Taylor expanding, we obtain:

$$\begin{aligned}\tau_m^n &= Au_{m-1}^n + Bu_m^n + Cu_{m+1}^n - \partial_{xx}u_m^n \\ &= (A+B+C)u + (shC - hA)u_x + \frac{1}{2}(Ah^2 + C(hs)^2 - 2)u_{xx} + \mathcal{O}(h^3).\end{aligned}$$

Solving for A, B, C such that the three lower order terms vanish gives the finite difference approximation:

$$\partial_{xx}u_m^n \approx \frac{1}{h^2}\left(\frac{2}{1+s}u_{m-1}^n - \frac{2}{s}u_m^n + \frac{2}{s(s+1)}u_{m+1}^n\right).$$

We remark that for $s = 1$ this coincides with the usual central difference. With a shrinking factor t in the northern direction, we can write down the full scheme as:

$$\nabla^2 u_m^n \approx \mathcal{L}_h U_m^n = \frac{1}{h^2}\left(\frac{2}{1+s}U_{m-1}^n + \frac{2}{s(s+1)}U_{m+1}^n + \frac{2}{1+t}U_m^{n-1} + \frac{2}{t(t+1)}U_m^{n+1} - \frac{2s+2t}{st}U_m^n\right).$$

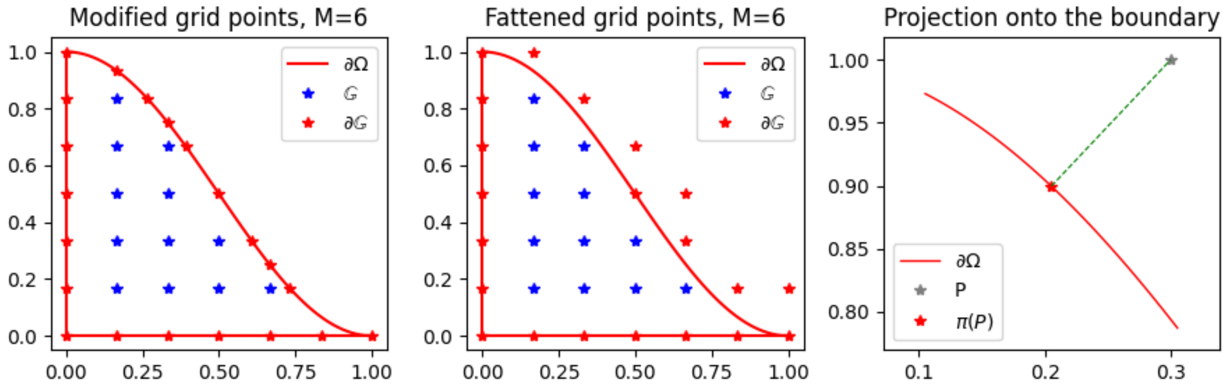
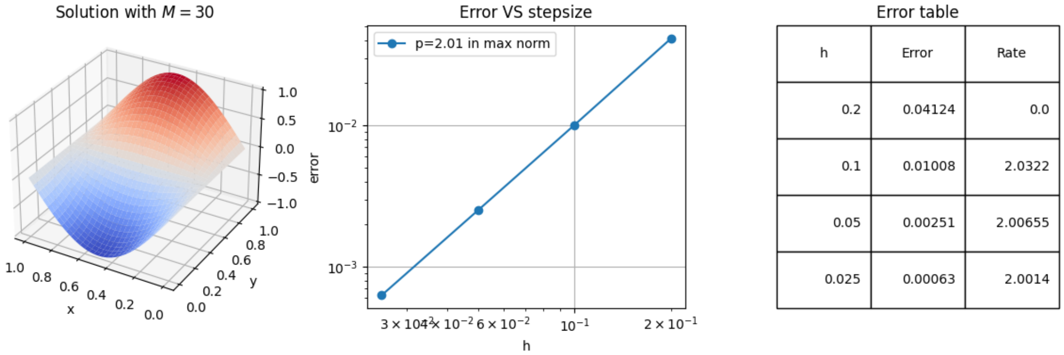


Figure 1: The boundary $\partial\Omega$ of the irregular domain, along with grid points for the two methods for handling the boundary. The rightmost plot show how the nearest boundary point can be found by projecting from the point $P \in \partial\mathbb{G}$.

3 Results and discussion

3.1 The $r = 1$ case

To test the convergence of the scheme we manufacture a solution and measure the error for different h. The results are seen in Figure 2. The slope agrees with the rate we found in Equation (4).



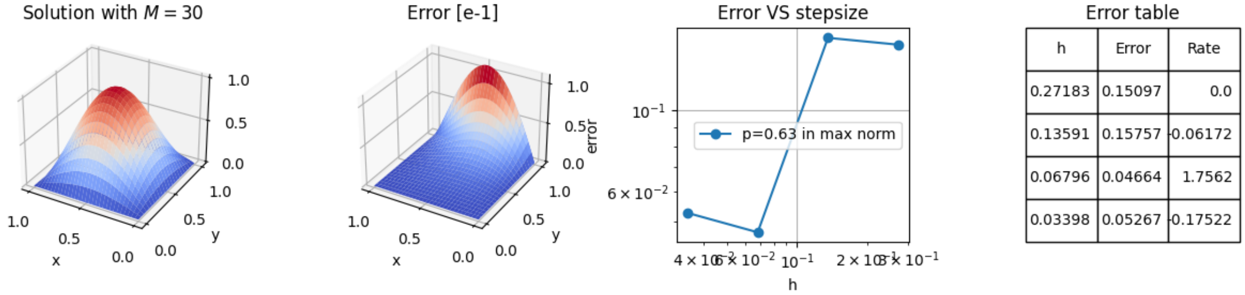
h	Error	Rate
0.2	0.04124	0.0
0.1	0.01008	2.0322
0.05	0.00251	2.00655
0.025	0.00063	2.0014

Figure 2: Results from error experiment for the manufactured solution $T(x, y) = \sin(\pi x)\cos(\pi y)$. The leftmost plot shows one solution. The middle graph is a log-log plot of the error and stepsize as given in the table.

Similar plots for different manufactured solutions can be found in Section 5. Here we find coinciding results to this, namely a convergence rate of 2, and solutions that look as expected.

3.2 Irrational r

We repeat the experiment above for testing convergence in the case where r is irrational, using both extension of the boundary condition and projection onto the boundary. The results for extended boundary conditions are shown in the appendix, as the results using projection are more interesting. These are shown in Figure 3.



h	Error	Rate
0.27183	0.15097	0.0
0.13591	0.15757	-0.06172
0.06796	0.04664	1.7562
0.03398	0.05267	-0.17522

Figure 3: Error experiment for the manufactured solution $T(x, y) = \sin(\pi x)\sin(\pi y)$ using $r = e/2$ and $a = 3$. The boundary at $y = 1$ is fattened and values are obtained by projection.

The second plot shows a significant increase in the error towards the fattened boundary at $y = 1$. This is somewhat as expected, since we are approximating the boundary conditions. From the third plot we also find that the error does not seem to strictly decrease with the step size h. We suspect this is dependent of how much the boundary is fattened, or in other words, by how much the grid misses the boundary at $y = 1$. This suggests that a good idea is to strategically choose step sizes that minimise how much one misses the boundary, but these experiments were outside the scope of our study.

3.3 Non-rectangular boundary

We present two variations of the fattened boundary together with the modified discretisation. Some results are shown in Figure 4. In the leftmost we assume that the boundary function g is defined at the external boundary points in the fattened boundary. This assumption makes implementation easy and gives the same error as the more involved modification of the boundary. It does however have the limitation that g must be defined outside $\partial\Omega$. From a theoretical perspective many functions have this issue, like the logarithm and the square root. From a practical standpoint, we may have measurements along the boundary of our object of study. Implementing the projection solving Equation (5) lets us use only the values on the boundary, but it also adds a $\mathcal{O}(h)$ error, which is reflected in the figure. Modifying the discretisation near the boundary is more work to implement, but it resolves both the issues encountered when fattening. We also remark that the error is less smooth near the irregular boundary. This could be due to the varying step sizes.

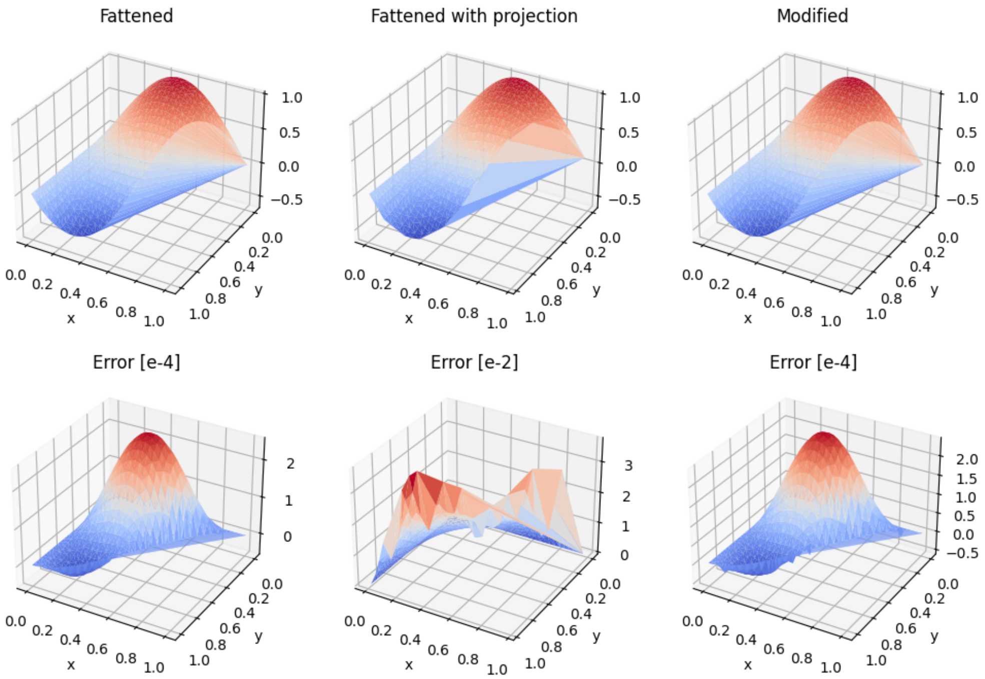


Figure 4: The numerical solution and error for the manufactured solution $T(x, y) = \sin(\pi x)\cos(\pi y)$. For the case of projection, the external boundary points are removed after computing solution.

4 Conclusion

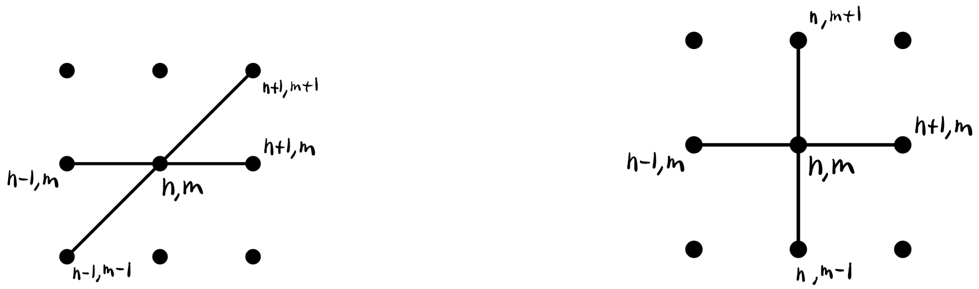
In this paper we researched the use of finite difference methods for numerically solving PDE's. In particular we have solved the anisotropic heat equation on rectangular domains and the isotropic heat equation on an irregular domain. We have found that missing the boundary of the domain when discretising might be an issue, which can be solved by either fattening the boundary or modifying the discretisation. If the boundary condition is defined just outside the boundary using those values gives good results. However if it is only defined exactly on the boundary one could project the points onto the boundary, but this leads to larger errors. Modifying the discretisation near the boundary avoids this, but can be more difficult to implement.

References

[1] Owren, B. *TMA4212 Numerical solution of partial differential equations with finite difference methods*. NTNU, 2017.

5 Appendix

5.1 Stencils

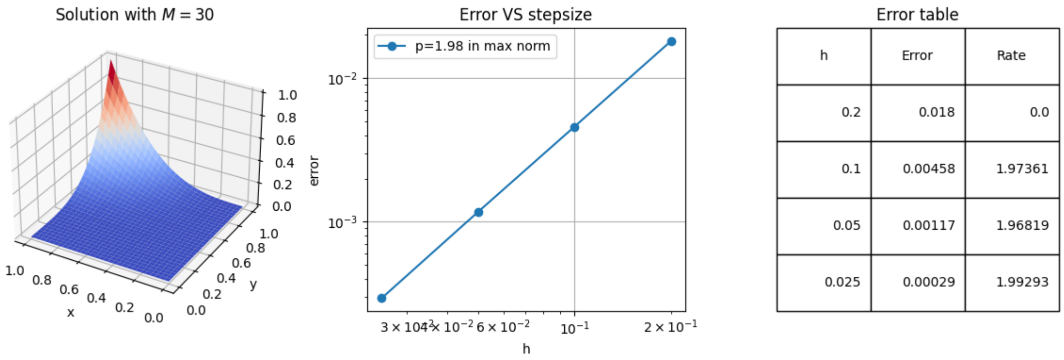


(a) Anisotropic case.

(b) Isotropic case

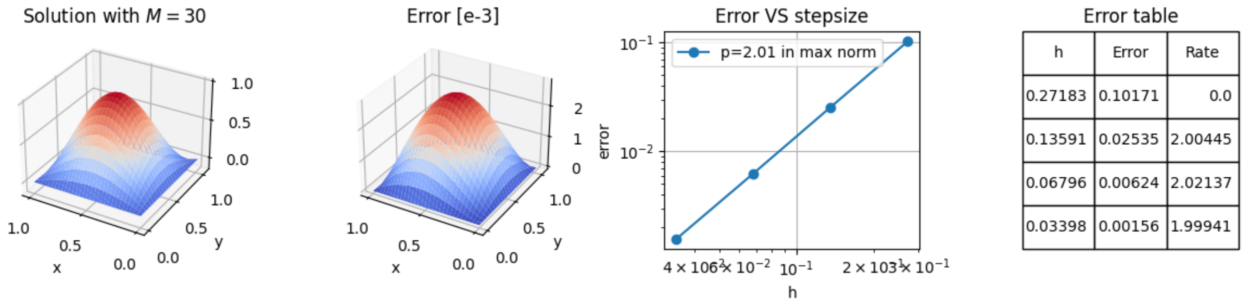
Figure 5: Stencils for the two discretisations.

5.2 Additional results



h	Error	Rate
0.2	0.018	0.0
0.1	0.00458	1.97361
0.05	0.00117	1.96819
0.025	0.00029	1.99293

Figure 6: The same experiment as in Figure 2, but with the solution $T(x, y) = x^3y^4$.



h	Error	Rate
0.27183	0.10171	0.0
0.13591	0.02535	2.00445
0.06796	0.00624	2.02137
0.03398	0.00156	1.99941

Figure 7: The same experiment as in Figure 3, but this time extending the boundary condition function instead of projecting.