# Numerical solution of temperature distributions in materials

August Rolfsen, Ine Daiwei Zhao, and Matias Sunde Øiesvold

February 2023

## 1 Introduction

This paper presents a solution to a stationary temperature distribution $T$ in anisotopic materials and isotropic materials given Dirichlet boundary conditions using finite difference meth6ods. For anisotropic materials, the solution is presented on a regular rectangular grid. For isotropic materials, the solution is presented on an irregular grid.

The Poisson equation is used to model stationary temperature distribution. If $\kappa$ is heat conductivity, and the solid has an internal heat source with energy density $f$, we can use conservation of energy, Fourier's law for heat flux, and the fact that we have a stationary problem to get the following model,

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

where $\Omega$ is the material solid and our domain.

The heat in anisotropic materials will by definition depend on the direction, which means the heat flows faster in some directions of the material. This means that $\kappa$ is a matrix. We will only consider two-dimensional models, with two directions for the heat flow $\vec{d_1} = (1, 0)$ and $\vec{d_2} = (1, r)$, with $r \in \mathbb{R}$

In the 2D case, $\kappa$ will be a $2 \times 2$ matrix,and the elements of the matrix are denoted in the following way,

$$\begin{pmatrix} a+1 & r \\ r & r^2 \end{pmatrix} \quad \text{where} \quad a > 0 \quad \text{and} \quad R := \frac{a}{|\vec{d_2}|^2} = \frac{a}{1+r^2}$$

where $a > 0$ and R is the relative strength of the conductivity in direction $\vec{d_1}$ versus $\vec{d_2}$.

By using directional derivatives and letting $T = u$, the matrix can be written in the following way, which gives us a nice expression for the model,

$$-\big(\nabla \cdot \begin{pmatrix} a+1 & r \\ r & r^2 \end{pmatrix} \nabla T\big) = -\big((a+1)\partial_x^2 u + 2r\partial_x\partial_y u + r^2\partial_y^2 u\big) = -\big(a\partial_x^2 u + (\vec{d_2} \cdot \nabla)^2 u\big)$$

## 2 The scheme

We will start with a simple rectangular domain $\Omega = [0, 1] \times [0, 2]$ , with Dirichlet boundary conditions: $u = g = \{g_E, g_N, g_W, g_S\}$ on the boundary $\partial\Omega$. Let $r = 2, a = 1$, and consider a grid with step sizes $h = \frac{1}{M}$ in the $x$ direction and k $= rh$ in the $y$ direction, with $M \in \mathbb{R}$. Using second order central differences in the directions $\vec{d_1}$ and $\vec{d_2}$, the term $a\partial_x^2 u$ has the following second order central difference discretisation.

$$a\frac{U(x+h, y) - 2U(x, y) + U(x-h, y)}{h^2} \tag{2}$$

For the directional derivative, we do not use the form with mixed derivatives, as discretised mixed derivatives are unstable. Instead we directly discretise the second order directional derivative using the method of undetermined coefficients, doing the a derivation for a general $r$.

$$AU(x+h, y+rh) + BU(x, y) + CU(x-h, y-rh) \approx \partial_x^2 u + 2r\partial_x\partial_y u + r^2\partial_y^2 u \tag{3}$$

By Taylor expansion on the three terms on the left hand side, and omitting the arguments from $U$, that is $U := U(x, y)$, we have

$$AU + AhU_x + ArhU_y + \frac{A}{2}h^2U_{xx} + Arh^2U_{xy} + \frac{A}{2}r^2h^2U_{yy} + \frac{C}{6}(h\partial_x + rh\partial_y)^3U + \frac{A}{24}(h\partial_x + rh\partial_y)^4U$$

$$O(h^5) + BU + CU - ChU_x - CrhU_y + \frac{C}{2}h^2U_{xx} + Ch^2rU_{xy} + \frac{C}{2}h^2r^2U_{yy}$$

$$-\frac{C}{6}(h\partial_x + rh\partial_y)^3U + \frac{C}{24}(-h\partial_x - rh\partial_y)^4U + O(h^5)$$

We need to impose constraints on the LHS constants in order for this to approximate the RHS of (3) We get the following set of equations, where many of them are dependent, and thus can be reduced to three equations.

$$\begin{cases} A+B+C=0 \\ Ah-Ch=0 \\ Arh-Crh=0 \\ \frac{A}{2}h^2+\frac{C}{2}h^2=-2r \\ \frac{2A}{2}rh^2+\frac{2C}{2}h^2r=-2r \\ \frac{A}{2}r^2h^2+\frac{C}{2}h^2r^2=-r^2 \end{cases} \implies \begin{cases} A+B+C=0 \\ A-C=0 \\ A+C=\frac{2}{h^2} \end{cases} \implies \begin{cases} A=\frac{1}{h^2} \\ B=\frac{-2}{h^2} \\ C=\frac{1}{h^2} \end{cases}$$

Combining this with the approximation for the $a\partial_x^2$ term, we get the following discretisation:

$$a\frac{U(x+h,y)-2U(x,y)+U(x-h,y)}{h^2}\frac{U(x+h,y+rh)-2U(x,y)+U(x-h,y-rh)}{h^2} \tag{4}$$

A diagram of the stencil in index-form can be seen in the appendix in figure (5). Writing this in matrix form as a system $A\vec{U}=\vec{F}+\vec{G}$, where

$$A=tridiag(B_0,B,B_1), B_0=tridiag(0,0,-1), B_1=tridiag(-1,0,0), B=tridiag(-1,4,-1)$$

with $A\in\mathbb{R}^{(M-1)^2\times(M-1)^2}, B_0,B,B_1\in\mathbb{R}^{(M-1)^2}, \vec{U},\vec{F},\vec{G}\in\mathbb{R}^{(M-1)}$ for a suitable $\vec{G}$ that takes care of boundary conditions. An example of the A matrix is given in the appendix, see figure 7.

## 2.1 Implementation and testing

We can now implement this scheme, by building the matrix, solving the system, and testing it on a suitable test solution. We propose two test problems: $u_1(x,y)=x^2+y^2$ and $u_2=cos(x)sin(y)$ with their corresponding right hand sides

$f_1=-(2(a+1)+2r^2)$
$f_2=(a+1)cos(x)sin(y)+2rsin(x)cos(y)+2r^2cos(x)sin(y)$

See figure (8) and (9) for convergence plots. From the convergence plot, we can see that this is a second order method. Note that the function $u_1$ is solved to within machine precision, which is why the convergence plot is not very insightful. Now, we will prove this so far empirically obtained result that our scheme has quadratic convergence.

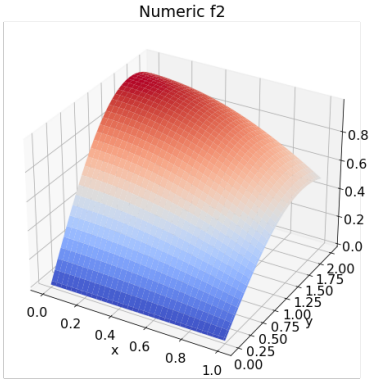


Figure 1: Numerical solution of $u_2$ with RHS $f_2$

## 2.2 Proving convergence

In order to prove convergence, we will have to prove stability and consistency. We will prove stability by first proving monotonicity.

**Definition**: *Monotonicity.* A scheme of the form $\mathcal{L}u=a_{i,i}u_{i,i}-\sum_{i\neq j}a_{i,j}u_{i,j}$ is monotone if there are positive coefficients $a_{i,j}>0$ for all $i,j$, and there is diagonal dominance $a_{i,i}\geq\sum_{i\neq j}|a_{i,j}|$.

### 2.2.1 Lemma 1: The scheme defined in equation (4) is monotone

Proof: We define the operator of the scheme in the following way,

$$-\mathcal{L}_hU_{m,n}=\frac{2(a+1)}{h^2}U_{m,n}-\frac{a}{h^2}U_{m-1,n}-\frac{a}{h^2}U_{m+1,n}-\frac{1}{h^2}U_{m-1,n-1}-\frac{1}{h^2}U_{m+1,n+1} \tag{5}$$

where $\mathcal{L}_h$ is the discrete operator with step size $h$.

From (5), there are positive coefficients and digaonal dominance in the scheme, which are

$$\frac{2(a+1)}{h^2}, \frac{a}{h^2}, \frac{1}{h^2} > 0 \tag{6}$$

$$a_{i,i} = \frac{2(a+1)}{h^2} \geq \sum_{i \neq j} |a_{i,j}| = \frac{a}{h^2} + \frac{a}{h^2} + \frac{1}{h^2} + \frac{1}{h^2} \quad a > 0, \quad h = \frac{1}{M}$$

Hence, the scheme satisfies these two conditions and we can conclude that the scheme is monotone.

### 2.3 Truncation error

Our truncation error is defined as follows:

$$\tau = u - U = (a\partial_x^2 u + \partial_x^2 u + 2r\partial_x\partial_y u + r^2\partial_y^2 u) - \\ a\frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} + \frac{u(x+h,y+rh) - 2u(x,y) + u(x-h,y-rh)}{h^2}$$

We will look at the two expressions separately and Taylor expand. The result for the first expression $a\partial_x^2 u$ is known from theory to be $\tau_1 = \frac{ah^2}{12}\partial_x^4 u + O(h^4)$. The second part is

$$\begin{aligned}
\tau_2 &= \partial_x^2 u + 2r\partial_x\partial_y u + r^2\partial_y^2 u - \frac{1}{h^2}\Bigg(u + (h\partial_x + rh\partial_y)u + \frac{1}{2}(h\partial_x + rh\partial_y)^2 u + \frac{1}{3!}(h\partial_x + rh\partial_y)^3 u \\
&+ \frac{1}{4!}(h\partial_x + rh\partial_y)^4 u + \frac{1}{5!}(h\partial_x + rh\partial_y)^5 u + O(h^6) - 2u + u - (h\partial_x + rh\partial_y)u + \frac{1}{2}(h\partial_x + rh\partial_y)^2 u \\
&- \frac{1}{3!}(h\partial_x + rh\partial_y)^3 u + \frac{1}{4!}(h\partial_x + rh\partial_y)^4 u - \frac{1}{5!}(h\partial_x + rh\partial_y)^5 u + O(h^6)\Bigg) \\
&= \frac{1}{12h^2}((h\partial_x + rh\partial_y)^4 u + O(h^6))
\end{aligned}$$

Where in both parts the odd terms in the Taylor expansion are canceled, along with various other cancellations. Putting this together we can formulate the truncation error in various different forms with directional derivatives and Taylor's Formula:

$$\begin{aligned}
\tau &= \frac{ah^2}{12}\partial_x^4 u + O(h^4) + \frac{1}{12h^2}((h\partial_x + rh\partial_y)^4 u + O(h^6)) \\
&= \frac{ah^2}{12}\partial_x^4 u(x + \xi_1 h, y + \xi_2 rh) + \frac{1}{12h^2}(h\partial_x + rh\partial_y)^4 u(x + \xi_3 h, y + \xi_4 rh), \quad \xi_i \in (0,1) \quad \forall \quad i \quad \in \quad [1,4] \\
&= \frac{ah^2}{12}\partial_x^4 u(x + \xi_1 h, y + \xi_2 rh) + \frac{1}{12h^2}(\vec{d_2} \cdot \nabla)^4 u(x + \xi_3 h, y + \xi_4 rh) \\
&= \frac{1}{12}\Bigg(ah^2\partial_x^4 u(x + \xi_1 h, y + \xi_2 rh) + \frac{1}{h^2}(\vec{d_2} \cdot \nabla)^4 u(x + \xi_3 h, y + \xi_4 rh)\Bigg) = O(h^2)
\end{aligned}$$

We see that $\tau \xrightarrow{h \to 0} 0$, so our scheme is consistent.

### 2.4 $L^\infty$ -stability

Before proving $L^\infty$ -stability for the scheme, we first state the following theorem,

**Theorem 1**. *Discrete Maximum Principle (DMP)*
Let $\mathcal{L}_h$ be the differential operator on the scheme.

$$\text{If} \quad \mathcal{L}_h U_P \leq 0, \forall P \in \mathbb{G} \quad \text{then,} \quad \max_{P \in \mathbb{G}} U_P \leq \max_{P \in \partial\mathbb{G}} U_p$$

Continuing on is proving the result of $L^\infty$ -stability. Since the scheme is monotone, we use Theorem 1 (DMP) and a super solution $\phi \geq 0$ such that

$$-\mathcal{L}_h \phi_P = 1 (\geq 0), \text{for all points } P \in \mathbb{G} \tag{7}$$

where $\mathbb{G}$ is the set of all interior grid points, and $\partial\mathbb{G}$ is the set of all boundary points.
For our Poisson problem (1), we define the following super solution

$$\phi_P = \frac{1}{2}(x - x^2), \quad \phi'_P = \frac{1}{2} - x, \quad \phi''_P = -1, \quad \max_{0 \leq x \leq 1} \phi = \phi(\frac{1}{2}) = \frac{1}{8} \tag{8}$$

Further, we define

$$V_p = \begin{cases} -\mathcal{L}_h V_p = f_p, , & \text{in } \mathbb{G} \\ V_p = 0, & \text{in } \partial\mathbb{G} \end{cases}$$

$$W_p := V_p - \|\vec{f}\|_\infty \phi_P \tag{9}$$

Then by definition,

$$\begin{aligned} -\mathcal{L}_h W_p &= -\mathcal{L}(V_p - \|\vec{f}\|_\infty) \\ &= f_p - \|\vec{f}\|_\infty \leq 0 \end{aligned}$$

From Theorem 1 (DMP), we get the following

$$\begin{aligned} & W_p \leq \max_{\partial\mathbb{G}} W_p = \max_{\partial\mathbb{G}} (V_p - \|\vec{f}\|_\infty \phi_P) \leq 0 \\ \implies & V_p \leq \|\vec{f}\|_\infty \phi_P \end{aligned}$$

The same approach can be proved for $-V_p, -f$, which gives us

$$-V_p \leq \| - \vec{f}\|_\infty \phi_P \tag{10}$$

Hence, from (9), (10), we can conclude

$$\max |V_p| \leq \max \|\vec{f}\|_\infty \phi_P \tag{11}$$

By (8) and (11), we have the stability result

$$\|\vec{V}\|_\infty \leq \frac{1}{8} \|\vec{f}\|_\infty \tag{12}$$

### 2.5 Error bound

The error equation is $e_P = u_P - U_P$ which gives the following

$$\begin{cases} -\mathcal{L}_h e_P = -\tau_P & , P \in \mathbb{G} \\ e_P = 0 & \text{on } \partial\mathbb{G} \end{cases}$$

By (12), we get the error bound,

$$\|\vec{e_P}\|_\infty \leq \frac{1}{8} \|\vec{\tau_P}\|_\infty \leq \frac{1}{96} \left( a h^2 \partial_x^4 u(x + \xi_1 h, y + \xi_2 r h) + \frac{1}{h^2} (\vec{d_2} \cdot \nabla)^4 u(x + \xi_3 h, y + \xi_4 r h) \right) \tag{13}$$

We see that we have quadratic convergence. □

From this expression it's easy to see that $u_1 = x^2 + y^2$ will be solved with machine precision, as $\partial_x^4 u_1 = \partial_y^4 u_1 = 0$.

### 2.6 Adapting the grid to let $r$ be irrational

By choosing $r = 2$, we had a nice enough value of $r$ that we were able to compute the directional derivative on our grid, but this is not always the case. To be able to hit the upper boundary on $\Omega = [0, 1] \times [0, 2]$, that would imply $r \cdot n = 2$, for some $n \in \mathbb{Q}^+$. Note that this is only possible if we allow $r$ to be rational, as $r = \frac{2}{n}$ is a rational number. This means that if we now use an irrational $r$, we certainly will not hit the boundary when $h = \frac{1}{M}, k = |r|h$.

In order to fix this issue, we let the closest grid point above our domain $[0, 2]$ in the y-direction to be evaluated as the boundary point, which introduces a linear error.

We will plot this for three interesting irrational values: $\frac{\pi}{2}$ and $\frac{2\pi}{\pi \pm \epsilon}$, $\epsilon = 10^{-10}$ The reason for choosing these values is that we will get two irrational numbers with values close to 2. For the value slightly larger than 2, we will barely step over the grid, so the error is practically negligible and therefore we still expect to have quadratic convergence. However, if we have a value slightly smaller than 2, we will have to make an extra step, and will get the largest possible error of about a whole step length $h$. Therefore, we expect linear convergence as the error on the boundary is linear. We are looking at the max error, and from figure 2, this results in convergence rates very near 2 and 1, which agrees with our expectations. However, if we choose $r = \frac{\pi}{2}$ or some other irrational number not related to grid size, we get a more irregular pattern. Regardless of how irregular the pattern might look when sampling 4 points, the error will not be larger than the error from using $\frac{\pi}{\pi+\epsilon}$. Plots of the solution and error for $f_2$ can be found in figure (10)

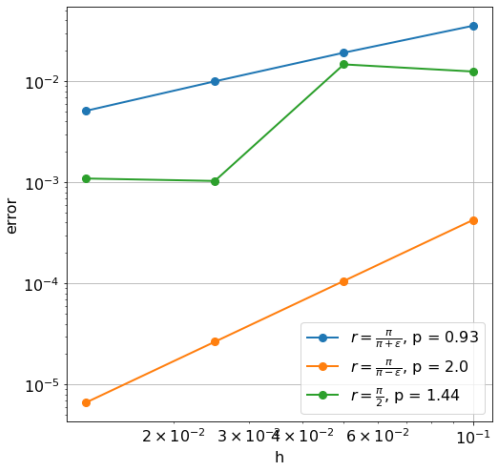


Figure 2: Convergence of $u_2$ with RHS $f_2$ for various values of $r$

## 3 Irregular grids

We will now consider an irregular domain

$$\Omega = \{(x, y) | x \geq 0, y \geq 0, y \leq 1 - x^2\}$$

For this problem, assume that the material is isotropic, such that $\kappa = I$, so we are solving the problem,

$$-\nabla^2 T = \Delta T = f$$

With an irregular domain, the grid also becomes irregular as the grid points will no longer align with the boundary points. Here we have implemented two different methods for solving this issue.

### 3.1 Adding the boundary nodes to the grid

The first method requires adding boundary points to the grid, which implies modifying the finite difference method near the boundary $\partial\Omega$. Since the boundary points lie in between two grid points, the distance is no longer the step size $h$ from an interior point $P$ to a boundary point. Here we will use a boundary point north $N'$. As a result,

$$\partial_y^2 u - \tau_{x,p} = a_{N'} u_{N'} + a_p u_p + a_s u_s$$

If we Taylor expand and use the method of undetermined coefficients we get

$$\begin{cases} a_{N'} + a_P + a_S = 0 \\ a_{N'}\eta_{N'}h - ha_S = 0 \\ \frac{1}{2}a_{N'}\eta_{N'}^2h^2 + a_s\frac{h^2}{2} \end{cases} \implies \begin{cases} a_{N'} = \dfrac{2}{h^2\eta_{N'}^2 + h^2\eta_{N'}} \\ a_P = -\dfrac{2}{h^2\eta_{N'}} \\ a_S = \dfrac{2}{h^2\eta_{N'} + h^2} \end{cases} \tag{14}$$

Note that when we implement these equations we need to change the signs. Similar results can be obtained for the other directions.

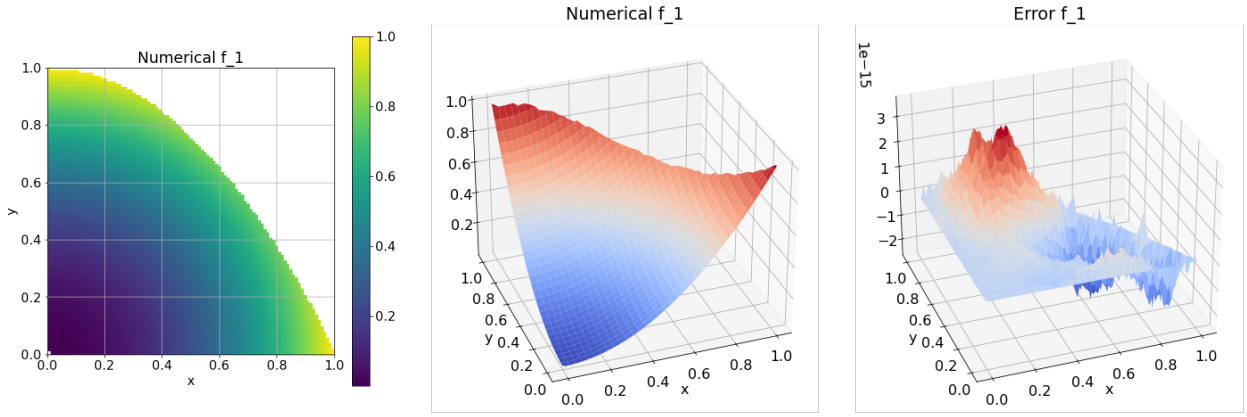


Figure 3: Heatmap, 3d plot and error plot on of $f_1$ on the irregular grid with added boundary nodes.

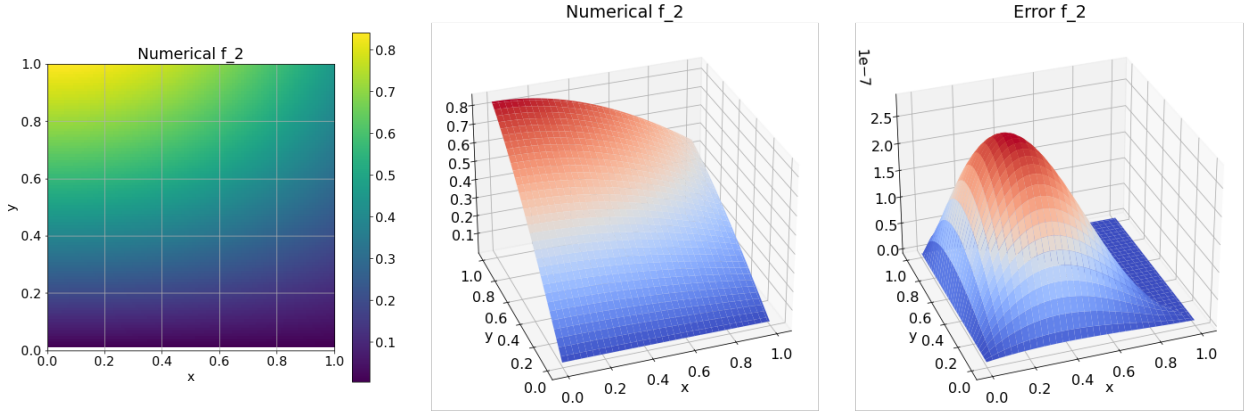


Figure 4: Plots for $f_2$, here with alternative visualization of the method of adding boundary nodes. For plotting purposes only, we have used the numerical solution for the irregular grid, but filled out the remaining outside nodes in the rectangle using the analytical solution. We see that the transition is smooth, which suggests that the method is working.

Since the grid points are individually modified and adjusted, the numerical error is very low. We have quadratic convergence, see figure (12) for a plot of $u_2$. $u_1$ is still machine precision, see figure (11) We did not expect to obtain quadratic convergence. For quadratic convergence to hold in general we would need $\frac{a_{N'}}{6}\eta_{N'}^3h^3 - \frac{a_S}{6}h^3 = 0$ in addition to the equations in (14), which we do not have enough free variables to satisfy.

### 3.2 Fattening the boundary

The second method is fattening the boundary, which consists of extending the boundary conditions to the closest grid points outside of the domain. The values for the boundary points are now calculated by projection of the nearest grid point outside of the domain onto the boundary.

Due to this assumption, there is a larger error along the boundary than the previous method. This is visible in figure 5 of $f_1 = x^2 + y^2$. $f_2$ is shown in figure (15) in the appendix. The method gives us linear convergence along the boundary, which results in overall linear convergence as our norm is considering the largest overall difference. A convergence plot is given in (13) and (14)

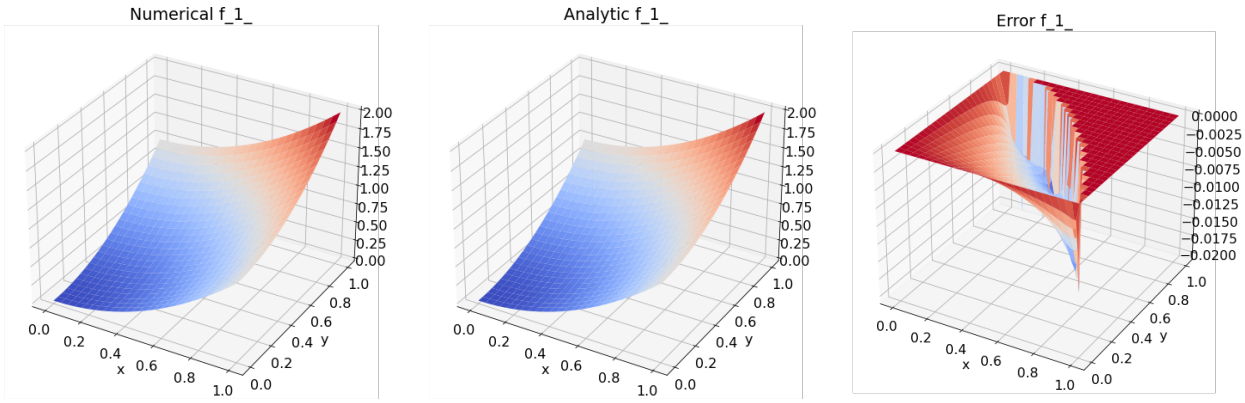


Figure 5: 3d plot of the numerical and analytic $f_1$ on the irregular grid when fattening the boundary. Again we have filled out the rest of the square grid as explained in figure (4). We see in the error plot that the error is not smooth, but the error is small enough that the transition is indistinguishable on the leftmost plot of the numerical solution

### 3.3 Observations

Discretising the irregular domain results in different numbers of interior grid points between the rows in the grid. The number of the interior grid points per row were calculated by $\lfloor \sqrt{(m*h+1)/h} \rfloor, m = 1, ..., M-1$ . With this function, there are certain cases of $M$ where the grid point could be a boundary point. Hence, by slightly adjusting the function to $\lfloor \sqrt{(m*h+1)/h - \zeta h} \rfloor, m = 1, ..., M-1$, where $\zeta < 10^{-6}$ or a small number, we are guaranteed that the boundary point is not included in the calculations. This makes sure that if we are very near a boundary point, we simply evaluate it, instead of fattening the boundary for that point.

It was significantly simpler to implement fattening of the boundary compared to altering the scheme. We tested runtime on the convergence calculations for $M = \{10, 20, 40, 80\}$. Altering the grid had an average wall time of approximately $0.9s$, while fattening the boundary had an average wall time of approximately $5s$. Altough slightly slower, the speed of the method is predominantly determined by $M$, so fattening the boundary is a good option as long as one has Dirichlet boundary conditions. Fattening the boundary presumably has a slower runtime as we need to solve the equation for the closest point on the boundary curve.

## 4 Summary

In this project we have implemented various different schemes for solving the Poisson equation on simple rectangular grids, rectangular grids with irrational directions, and irregular grids. Although cumbersome to implement, the final results were good. Finite element methods could be worth considering for these irregular grid problems, as long as the PDE is not too complicated.

## 5 Appendix

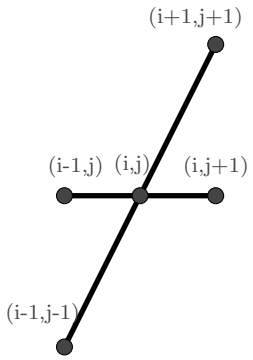


Figure 6: Stencil with index-notation

$$\begin{bmatrix} B & B_1 & 0 & 0 \\ B_0 & B & B_1 & 0 \\ 0 & B_0 & B & B_1 \\ 0 & 0 & B_0 & B \end{bmatrix}$$

Figure 7: Example of Matrix in task 1

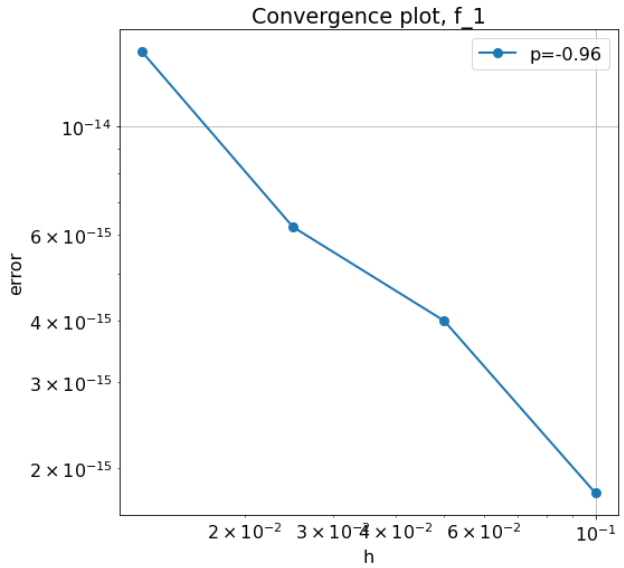


Figure 8: Convergence of $u_1$ with RHS $f_1$ on a regular grid. Notice the machine precision which explains the strange shape.

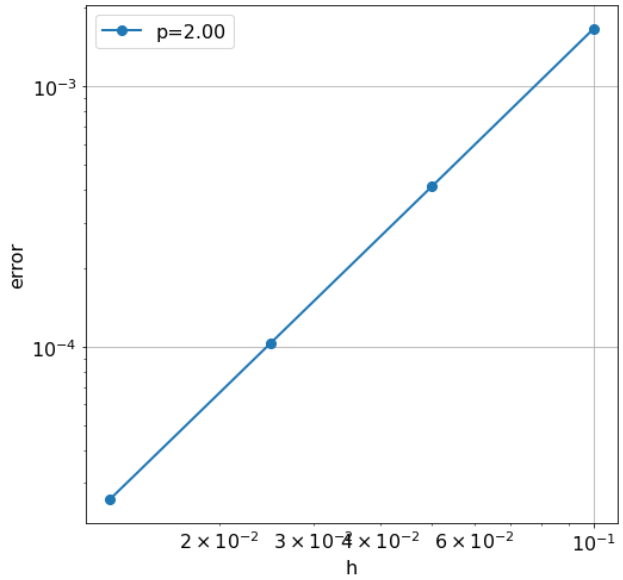


Figure 9: Convergence of $u_2$ with RHS $f_2$ on a regular grid

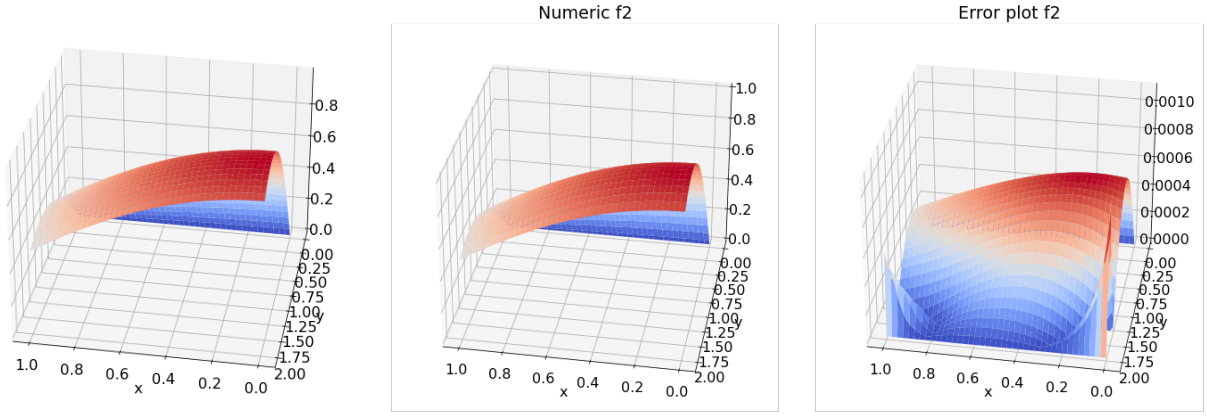


Figure 10: Analytic, numerical solution, and error plot of $f_2 = cos(x)sin(y)$ on the regular grid with $r = \frac{\pi}{2}$. Note that the error image is from a rotated point of view.

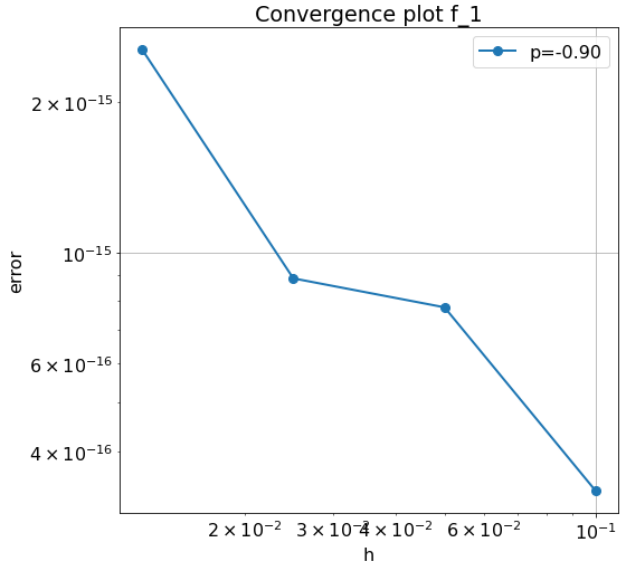


Figure 11: Convergence of $u_1$ on irregular grid with added boundary nodes. Still machine precision

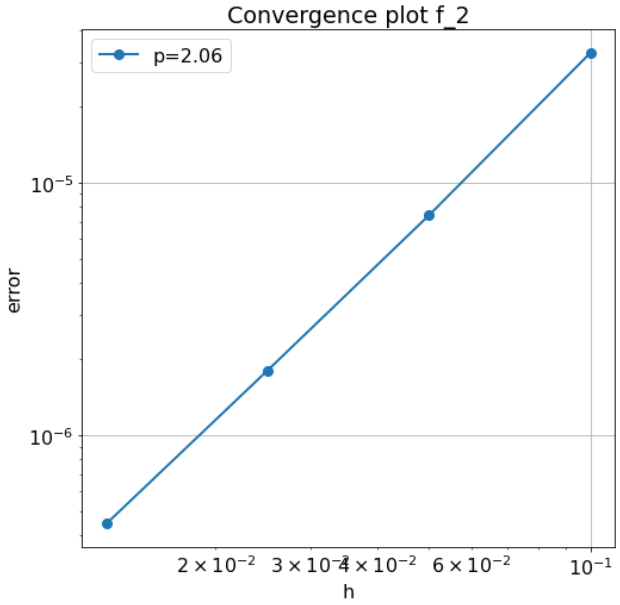


Figure 12: Convergence of $u_2$ on irregular grid with added boundary nodes, still quadratic convergence.

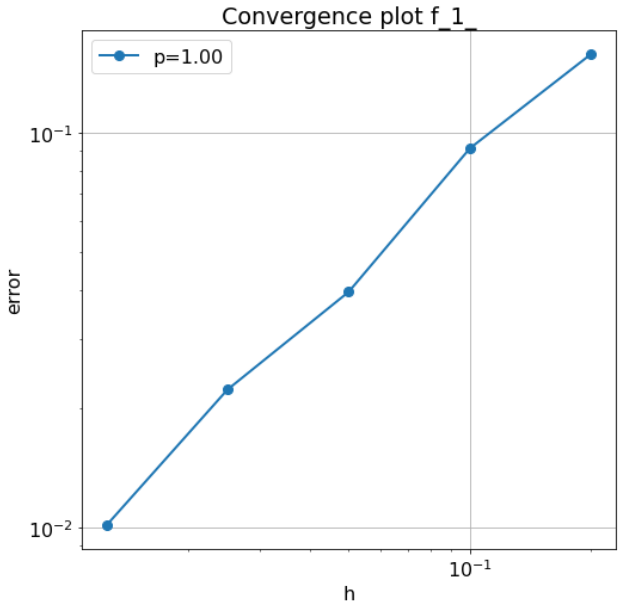


Figure 13: Convergence of $u_1 = x^2 + y^2$ on irregular grid with fattened boundary. Notice that we no longer have quadratic convergence. The fact that $p$ is so perfectly 1.00 is a bit of a coincidence. We see that the points are not on a perfect straight line, so it's an artifact of the choice to use linear regression.

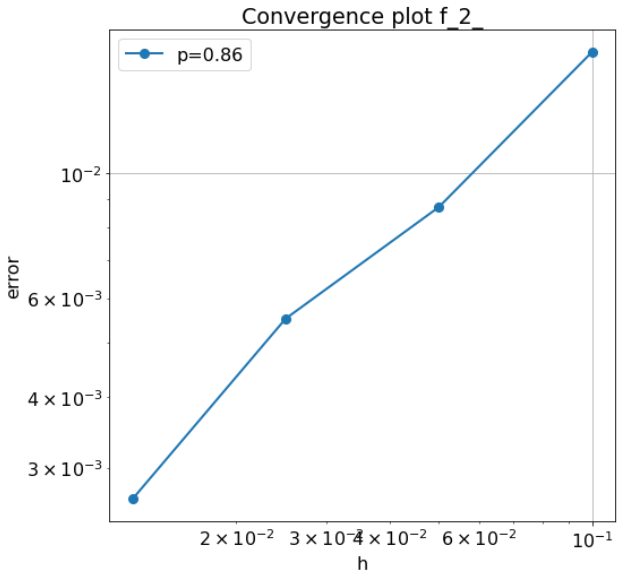


Figure 14: Convergence of $u_2$ on irregular grid with fattened boundary. We see that the line is not as smooth as the method of altering the scheme, this is due to the irregular error shape along the boundary in the error plots for fattening the boundary.

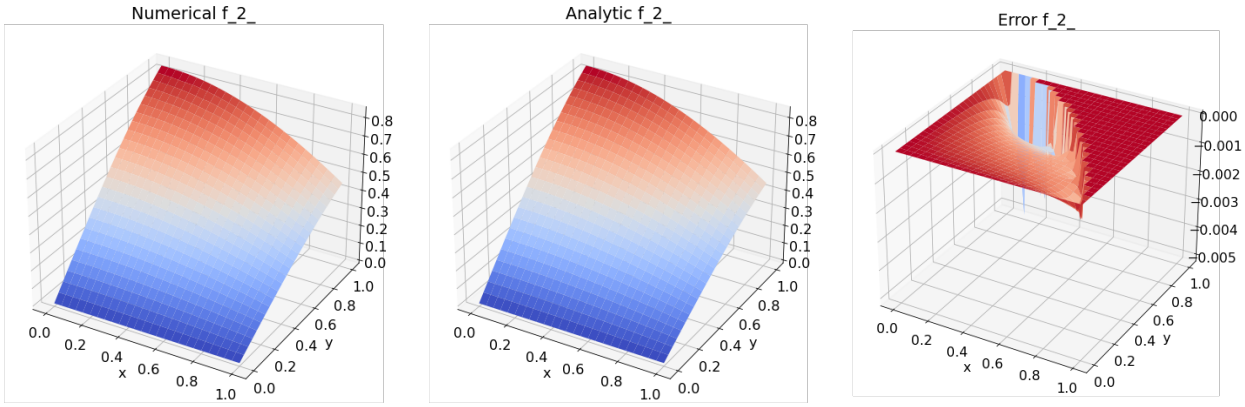


Figure 15: 3d plot of the numerical and analytic $f_2 = cos(x)sin(y)$ on the irregular grid when fattening the boundary. Again we have filled out the rest of the square grid as explained in (4)