

# Numerics Exercise 3

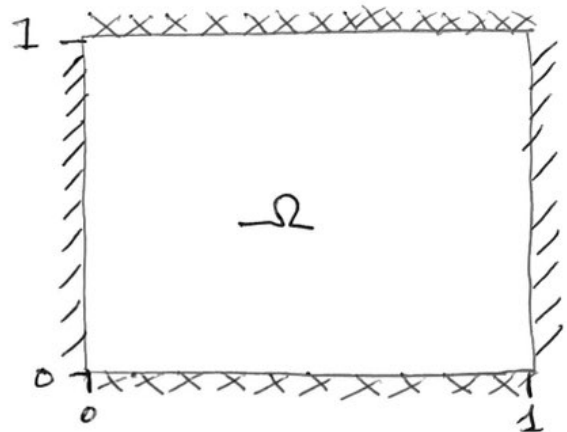
i)

/// = Dirichlet boundary conditions on walls  $\partial\Omega_D$

xxx = Neumann boundary conditions on walls  $\partial\Omega_N$

Dirichlet BC:  $u = 0$  on  $\partial\Omega_D$

Neumann BC:  $\frac{\partial u}{\partial y} = 0$  on  $\partial\Omega_N$



Poisson equation:  $-\nabla^2 u = f(x, y)$

Domain:  $\Omega \in [0, 1]^2$

Imposed force:  $f(x, y) = 2\pi^2 \sin(\pi x) \cos(\pi y)$

Exact solution:  $u_e(x, y) = \sin(\pi x) \cos(\pi y)$

To verify this:

$$\begin{aligned}\nabla^2 u_e &= \partial_x^2 (\sin(\pi x) \cos(\pi y)) + \partial_y^2 (\sin(\pi x) \cos(\pi y)) \\ &= -\pi^2 u_e - \pi^2 u_e \\ &= -2\pi^2 u_e \\ &= -2\pi^2 \sin(\pi x) \cos(\pi y) \text{ as expected} \\ &= -f\end{aligned}$$

Ritz Galerkin principle:

for the energy functional:  $J[u] = \iint_{\Omega} \frac{1}{2} (\nabla u)^2 - f(x, y) u \, dx \, dy$

we find our solution when  $\delta J[u] = J[u + \epsilon \delta u] - J[u] = 0$

ie when the ~~potential~~ energy of the system is minimised

i continued)

$$\delta J[u] = J[u + \epsilon \delta u] - J[u]$$

$$= \iint_{\Omega} \frac{1}{2} (\nabla(u + \epsilon \delta u))^2 - f(x, y) (u + \epsilon \delta u) dx dy \\ - \left( \iint_{\Omega} \frac{1}{2} (\nabla u)^2 - f(x, y) u dx dy \right)$$

$$= \iint_{\Omega} \frac{1}{2} (\nabla u^2 + 2 \nabla(\epsilon \delta u) \cdot \nabla u + \epsilon^2 (\nabla \delta u)^2 - (u^2) - \epsilon f \delta u dx dy$$

$$= \iint_{\Omega} \frac{\epsilon \nabla \delta u \cdot \nabla u}{2} + \frac{\epsilon^2 (\nabla \delta u)^2}{2} - \epsilon f \delta u dx dy$$

$$\frac{\delta J[u]}{\delta \epsilon} = \iint_{\Omega} \frac{\nabla(\delta u) \cdot \nabla u}{2} + \epsilon (\nabla \delta u)^2 - f \delta u dx dy$$

in  $\lim_{\epsilon \rightarrow 0}$

$$= \iint_{\Omega} \nabla(\delta u) \cdot \nabla u - f \delta u dx dy$$

$\therefore$  we have obtained:

$$\iint_{\Omega} \nabla(\delta u) \cdot \nabla u - f(x, y) \delta u dx dy = 0$$

Continuous  
Ritz Galerkin  
Formulation

Conditions on variable  $\delta u$ :  $\delta u(x, y)$  must be a smooth function within the domain  $\Omega$  so that  $\nabla(\delta u)$  is well defined

$\delta u$  must also be square integrable i.e.

$\iint_{\Omega} \delta u(x, y) dx dy$  must give a finite solution

3) 1 continued )

## Weak Formulation

$$-\nabla^2 u = f \quad \text{in } \Omega \in [0, T]^2$$

$$u = 0 \quad \text{on } \partial\Omega_D$$

$$\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$$

$$\nabla u \cdot n = 0 \quad \text{on } \partial\Omega_N$$

$$-\langle \nabla^2 u, w \rangle = \langle f, w \rangle$$

$$-\langle \nabla^2 u, v \rangle = \langle f, v \rangle \quad \text{where } v \text{ is chosen as an arbitrary test function that satisfies the boundary conditions}$$

$$\langle u, v \rangle = \iint_{\Omega} u(x, y) v(x, y) dx dy$$

Definition of the inner product

$\therefore$  we get:

$$-\iint_{\Omega} (u_{xx} + u_{yy}) v(x, y) dx dy = \iint_{\Omega} f(x, y) v(x, y) dx dy$$

$$\text{LHS: } -\iint_{\Omega} (u_{xx} + u_{yy}) v(x, y) dx dy$$

$$\text{we know: } \partial_x (u_x v) = u_{xx} v + u_x v_x$$

$$\partial_y (u_y v) = u_{yy} v + u_y v_y$$

rearrange (

$$u_{yy} v = \partial_y (u_y v) - u_y v_y$$

$$u_{xx} v = \partial_x (u_x v) - u_x v_x$$

} sub into integral

$$-\iint_{\Omega} (u_{xx} + u_{yy}) v dx dy = -\iint_{\Omega} \partial_y (u_y v) + \partial_x (u_x v) dx dy \quad \text{--- (A)}$$

$$+ \iint_{\Omega} (u_y v_y + u_x v_x) dx dy \quad \text{--- (B)}$$

1 continued)

to solve (A): apply green's theorem

$$\int_{\partial\Omega} P dx + Q dy = \iint_{\Omega} (Q_x - P_y) dy dx$$

$$\begin{aligned} - \iint_{\Omega} \left[ \overbrace{\partial_y(u_y v)}^{-P_y} + \overbrace{\partial_x(u_x v)}^{+Q_x} \right] dx dy &= - \int_{\partial\Omega} u_x v dy - u_y v dx \\ &= \int_{\partial\Omega_N} u_y v dx - \int_{\partial\Omega_D} u_x v dy \end{aligned}$$

we can choose  $v$  (as it is arbitrary) such that

$$\int_{\partial\Omega_D} u_x v dy = 0$$

in addition: we know that  $u_y = 0$  on  $\partial\Omega_N$

$\therefore$  we get: (A) = 0

$\therefore$  our weak formulation is:

$$\iint_{\Omega} [u_y v_y + u_x v_x] dx dy = \iint_{\Omega} f(x,y) v(x,y) dx dy$$

$$\text{in 2D: } u_y v_y + u_x v_x = \nabla u \cdot \nabla v$$

$\therefore$  we can write this as:

$$\iint_{\Omega} \nabla u \cdot \nabla v - f v \, dx dy = 0$$

continuous  
Weak formulation

where  $\Omega$ :  $0 \leq y \leq 1$ ,  $0 \leq x \leq 1$ ,  $f(x,y) = 2\pi \sin(\pi x) \cos(\pi y)$

1 continued)

We can see that the two must be equivalent by looking at comparing the ~~the two integrals~~ Continuous Ritz Galerkin formulation and the ~~weak~~ continuous weak formulation.

$$\iint_{\Omega} \nabla(\delta u) \cdot \nabla u - f(x,y) \delta u \, dx \, dy = 0 \quad \text{RG form}$$

$$\iint_{\Omega} \nabla v \cdot \nabla u - f(x,y) v \, dx \, dy = 0 \quad \text{Weak form}$$

both must be :

- smooth so that  $\nabla v / \nabla(\delta u)$  is well defined
- square integrable so  $\iint_{\Omega} v \, dx \, dy$  is finite

as  $v$  is an arbitrary test function (as is  $\delta u$ ) and the two forms share the same integral formulation, it is clear that  $v$  and  $\delta u$  are equivalent. They are both test functions that perturb the system and belong to a specific Hilbert space, specified by the boundary conditions.

## 2) Finding discrete Ritz Galerkin Formulation

from before:  ~~$J[u] = \iint_{\Omega} \sum_i \sum_j$~~

NB:  $u_j$  and  $v_j$  are interchangeable

$$J[u] = \iint_{\Omega} \frac{1}{2} |\nabla u|^2 - f(x,y) u \, dx \, dy$$

let:  $u \approx u_n \approx \sum_{j=1}^{N_n} u_j \psi_j(\bar{x})$

$$\iint_{\Omega} \frac{1}{2} (\nabla(\sum_i u_i \psi_i) \cdot \nabla(\sum_j u_j \psi_j)) - f u_j$$

$$- \sum_j f u_j \psi_j \, dx \, dy = 0$$

$$\begin{aligned} \nabla(u_i \psi_i) &= \underbrace{\psi_i \nabla u_i}_{=0} + u_i \nabla \psi_i \\ &= u_i \nabla \psi_i \end{aligned}$$

$u_j$ : coefficients

$\psi_j$ : basis function

$N_n$ : number of nodes where solution is unknown

$$N_T = N_n + N_K$$

↑  
total # of nodes

↖ # of known solution nodes (nodes on boundary  $\partial\Omega_0$ )

2 continued)

$$\therefore \iint_{\Omega} \frac{1}{2} \left( \sum_i \sum_j u_i \cancel{\varphi_i} + u_i \nabla \varphi_i, u_j \nabla \varphi_j \right) - \sum_j u_j f \varphi_j \, dx dy = 0$$

$$\therefore I[u] = \iint_{\Omega} \sum_i \sum_j \frac{u_i \nabla \varphi_i, u_j \nabla \varphi_j}{2} - \sum_j u_j f \varphi_j \, dx dy \quad \text{is the discretised Ritz gallerkin functional}$$

$$= \sum_i \sum_j \frac{A_{ij} u_i u_j}{2} - \sum_j b_j u_j$$

where we have defined:  $A_{ij} = \iint_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, dx dy$

$$b_j = \iint_{\Omega} f \varphi_j \, dx dy$$

it is convenient not to absorb factor of  $1/2$  into  $A_{ij}$

$$\delta I[u] = I[u + \epsilon \delta u] - I[u]$$

here:  $\left. \begin{array}{l} u_i \rightarrow u_i + \epsilon \delta u_i \\ u_j \rightarrow u_j + \epsilon \delta u_j \end{array} \right\} \text{ for } I[u + \epsilon \delta u]$

Performing a variation on  $\delta I[u]$

$$\therefore \delta I[u] = \sum_i \sum_j \frac{A_{ij}}{2} (u_i + \epsilon \delta u_i)(u_j + \epsilon \delta u_j) - \sum_j b_j (u_j + \epsilon \delta u_j)$$

$$- \sum_i \sum_j \frac{A_{ij} u_i u_j}{2} + \sum_j b_j u_j$$

$$= \sum_i \sum_j \frac{A_{ij}}{2} (u_i u_j + u_i \epsilon \delta u_j + \epsilon \delta u_i u_j + \epsilon^2 \delta u_j \delta u_i - u_i u_j) - \sum_j b_j \epsilon \delta u_j$$

$$= \sum_i \sum_j \frac{A_{ij}}{2} (\epsilon (u_i \delta u_j + \delta u_i u_j) + \epsilon^2 \delta u_j \delta u_i) - \sum_j b_j \epsilon \delta u_j$$

$$\frac{\delta I[u]}{\delta \epsilon} = \sum_i \sum_j \frac{A_{ij}}{2} (u_i \delta u_i + \delta u_i u_j + 2 \epsilon \delta u_j \delta u_i) - \sum_j b_j \delta u_j$$

let  $\epsilon \rightarrow 0$

$$= \sum_j \sum_i \frac{A_{ij}}{2} (u_i \delta u_j + u_j \delta u_i) - b_j \delta u_j$$

2)

2 continued)

since we defined:  $A_{ij} = \iint_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, dx dy$

it is clear to see that  $A_{ij} = A_{ji}$  (symmetric matrix)

$\therefore$  we can swap  $i \leftrightarrow j$  indices

$$\begin{aligned} \therefore \frac{\delta I[u]}{\delta \varepsilon} &= \sum_j \sum_i \frac{A_{ij}}{2} (u_j \delta u_i + u_i \delta u_j) - b_j \delta u_j \\ &= \sum_j \sum_i A_{ji} u_j \delta u_i - b_j \delta u_j = 0 \end{aligned}$$

$$\sum_j \sum_i A_{ji} u_j \delta u_i - b_j \delta u_j = 0$$

$$\sum_j \sum_i (A_{ji} u_j - b_j) \delta u_i = 0$$

Writing in einstein notation:

$$A_{ji} u_j - b_j = 0$$

$$A_{ij} u_j - b_i = 0$$

result when variational principle applied to discretised ritz-galerkin functional

### Discretised Weak Formulation

Continuous weak form:  $\iint_{\Omega} \nabla u \cdot \nabla v \, dx dy = \iint_{\Omega} f v \, dx dy$

let:  $u(x) \approx u_n(x) = \sum_{j=1}^{N_n} u_j \varphi_j$

where  $\varphi_j$  is <sup>a</sup> square integrable basis function

$\sum_i \sum_j \iint_{\Omega} \nabla(u_j \varphi_j) \cdot \nabla v \, dx dy - \iint_{\Omega} f v \, dx dy = 0$  let  $v = \varphi_i$  ie we choose our test function to be the same as our basis function

$$\sum_i \sum_j \iint_{\Omega} \nabla(u_j \varphi_j) \cdot \nabla \varphi_i - f \varphi_i \, dx dy = 0$$

This works as long as the basis functions are smooth and, square integrable and satisfy the boundary conditions

2 continued)

$$\sum_i \sum_j \iint_{\Omega} u_j \nabla \varphi_j \cdot \nabla \varphi_i - f \varphi_i \, dx \, dy = 0$$

let us define:  $A_{ij} = \iint_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, d\Omega$   $d\Omega = dx \, dy$

$$b_i = \iint_{\Omega} f \varphi_i \, d\Omega$$

$$\sum_i \sum_j A_{ij} u_j - b_i = 0$$



Einstein summation notation:  $A_{ij} u_j - b_i = 0$

This is the same expression that we obtained from applying the variational principle to the discretised Ritz-Galerkin Formulation,

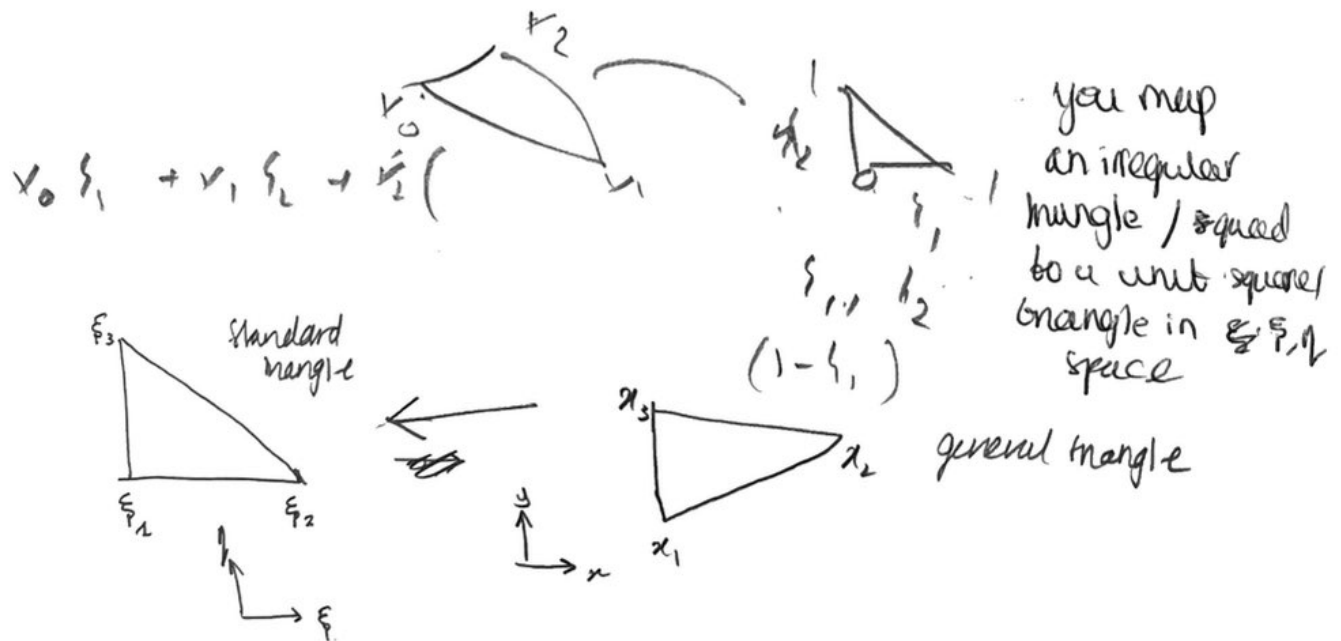
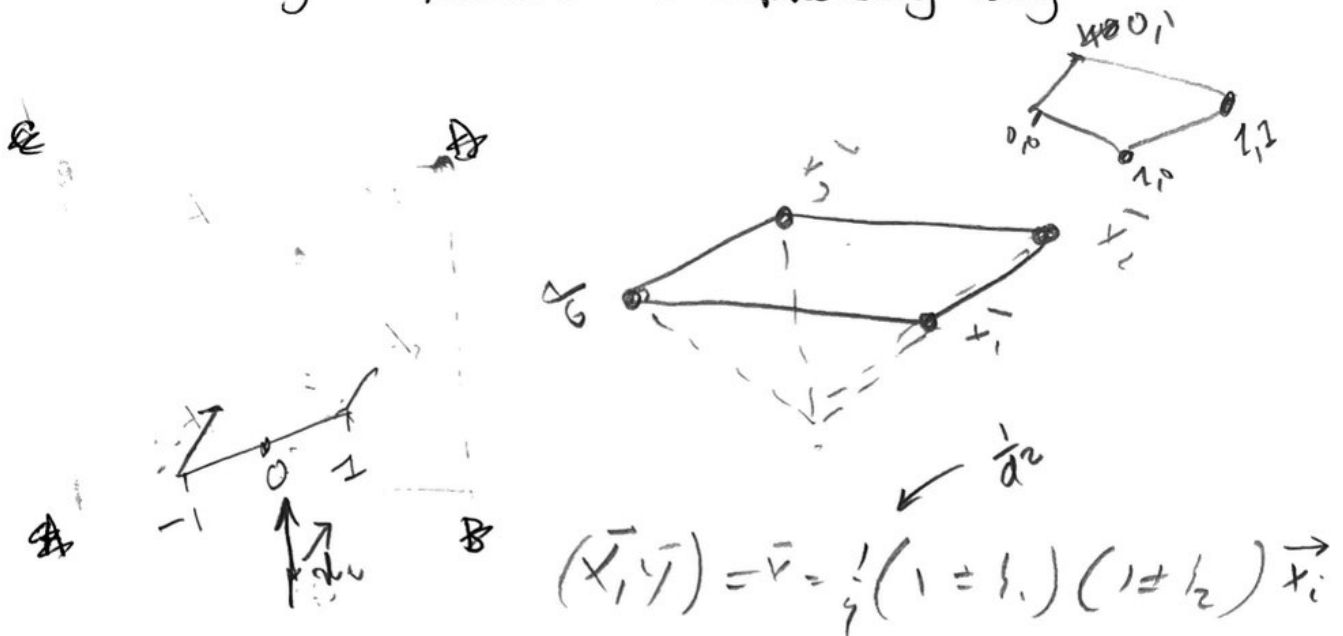
NB:  $A_{ij}$  is not a square matrix as  $u$  is known on  $\partial\Omega_0$  but not on  $\partial\Omega_N$ . We can rewrite it in the following way to turn it into a square matrix by separating out the Neumann entries (ie those on  $\partial\Omega_N$ )

$$A_{ik} u_k - b_i = - \sum_{m=N_0+1}^{N_T} A_{im} u_m$$

in the case of the Poisson equation, I think the RHS disappears?



3) We can use a barycentric coordinate system that expresses the position of a point P within ~~a~~ a rectangular element by ~~the following~~ in the following way



$$x_1 \rightarrow (0, 0) \quad x_2 \rightarrow (1, 0) \quad x_3 \rightarrow (0, 1)$$

basis functions:  $\phi_1(\xi, \eta) = 1 - \xi - \eta$

$$\phi_2(\xi, \eta) = \xi$$

$$\phi_3(\xi, \eta) = \eta$$

$$x = x_1 + (x_2 - x_1) \xi + (x_3 - x_1) \eta$$

$$y = y_1 + (y_2 - y_1) \xi + (y_3 - y_1) \eta$$

4)

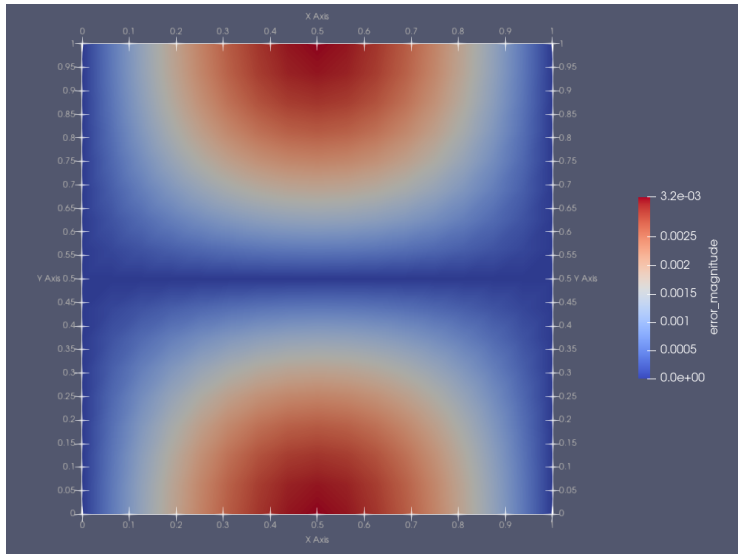
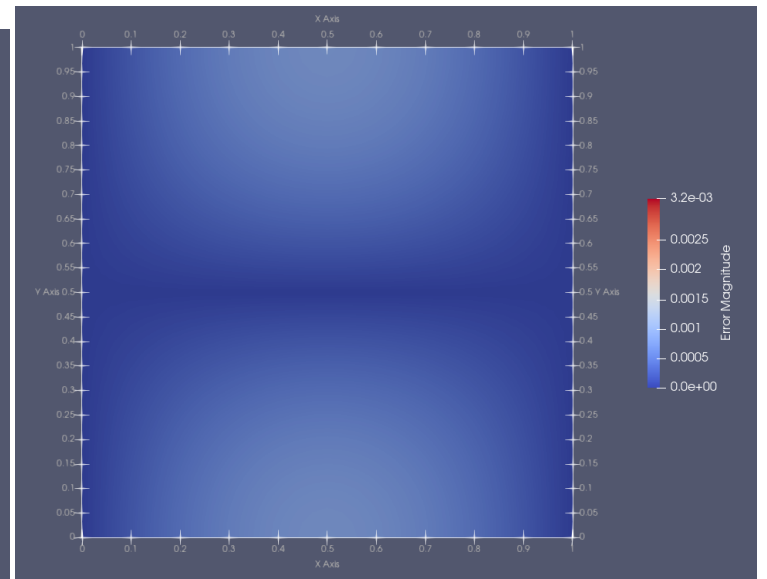
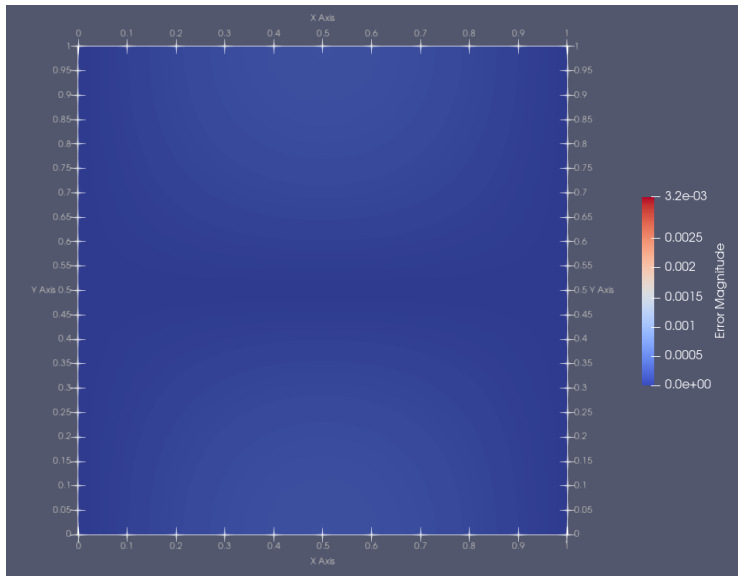
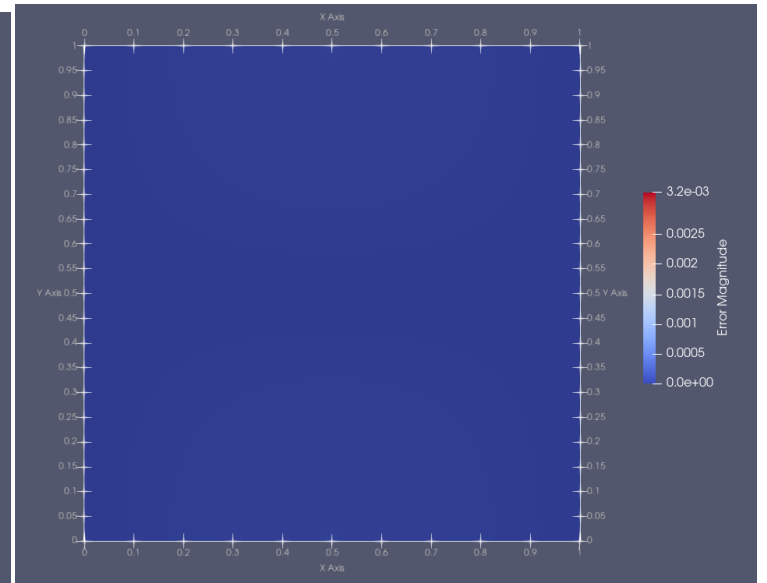
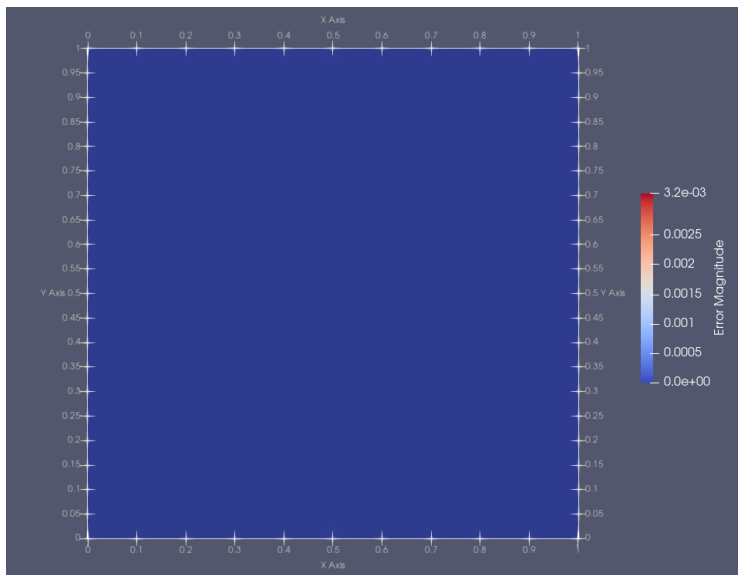
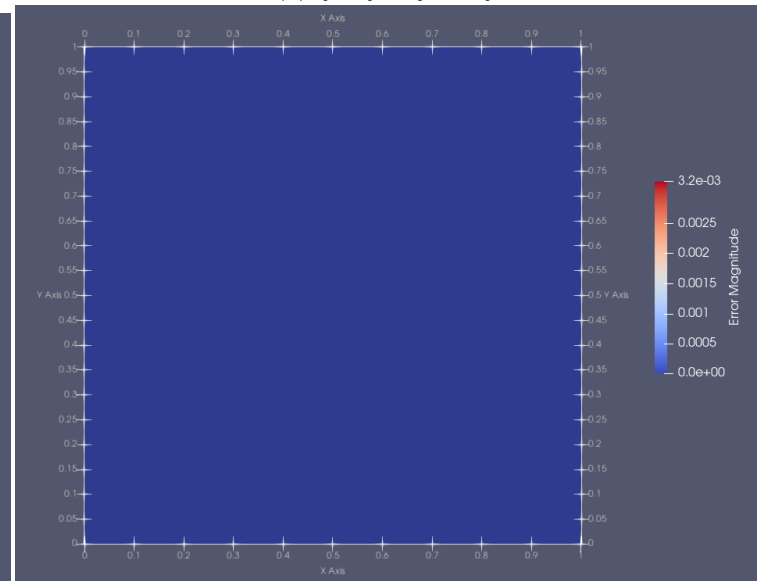
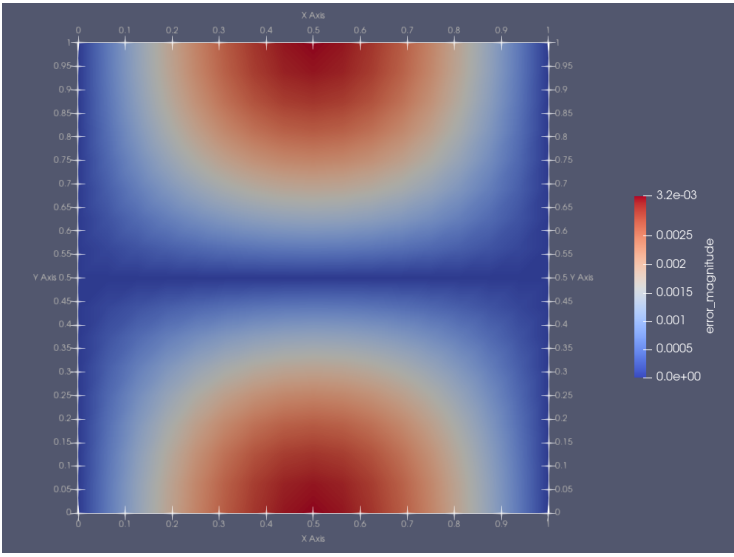
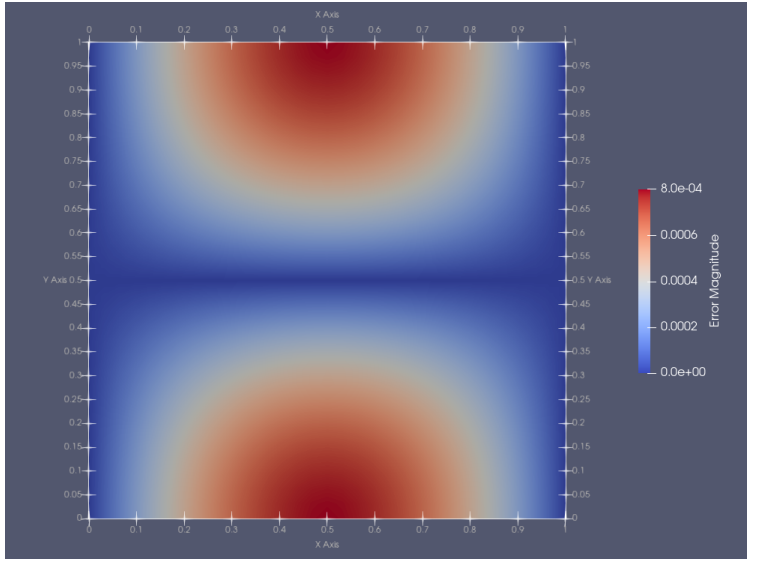
(a)  $\{h, p\} = \{16, 1\}$ (b)  $\{h, p\} = \{32, 1\}$ (c)  $\{h, p\} = \{64, 1\}$ (d)  $\{h, p\} = \{128, 1\}$ (e)  $\{h, p\} = \{256, 1\}$ (f)  $\{h, p\} = \{512, 1\}$ 

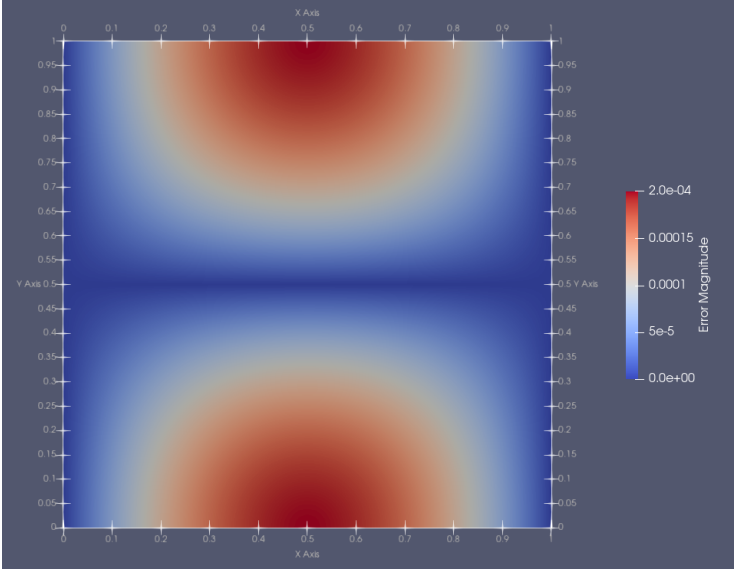
Figure 1: A figure showing how  $|u_h - u_e|$  changes as  $h$  increases, where  $u_h$  is the numerical approximation and  $u_e$  is the exact solution. The scale on each colour map remains the same, highlighting that the error associated with discretising the Poisson equation using a 1st order FEM is minimised when the resolution of the mesh increases.



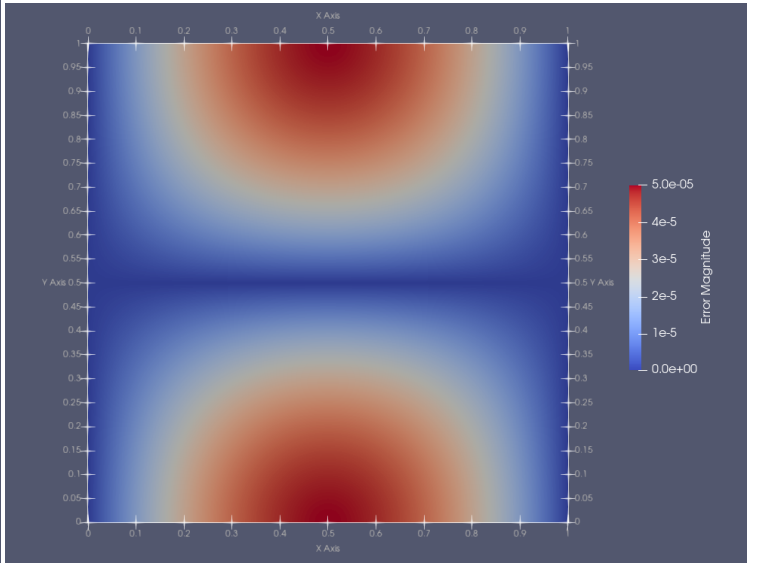
(a)  $\{h, p\} = \{16, 1\}$



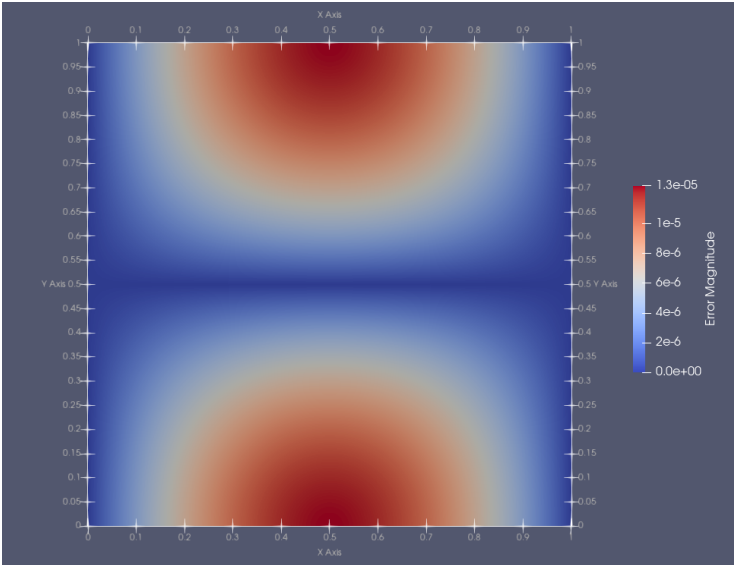
(b)  $\{h, p\} = \{32, 1\}$



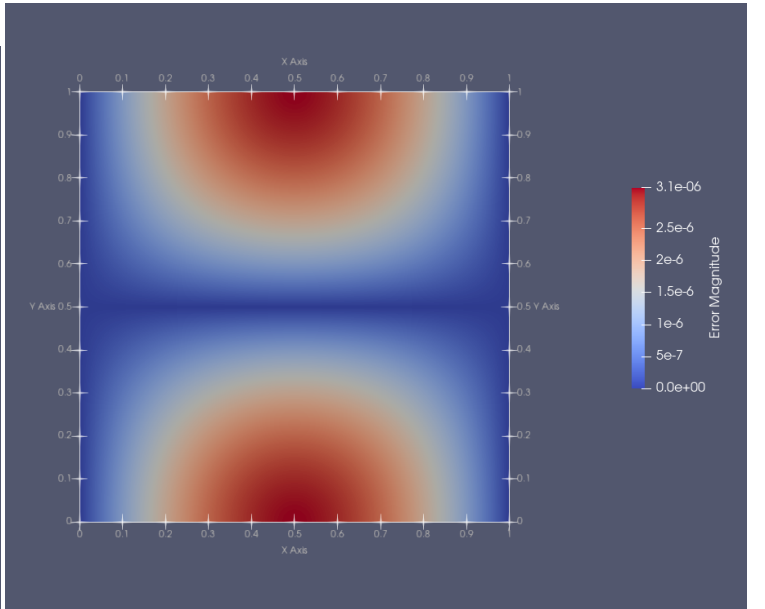
(c)  $\{h, p\} = \{64, 1\}$



(d)  $\{h, p\} = \{128, 1\}$

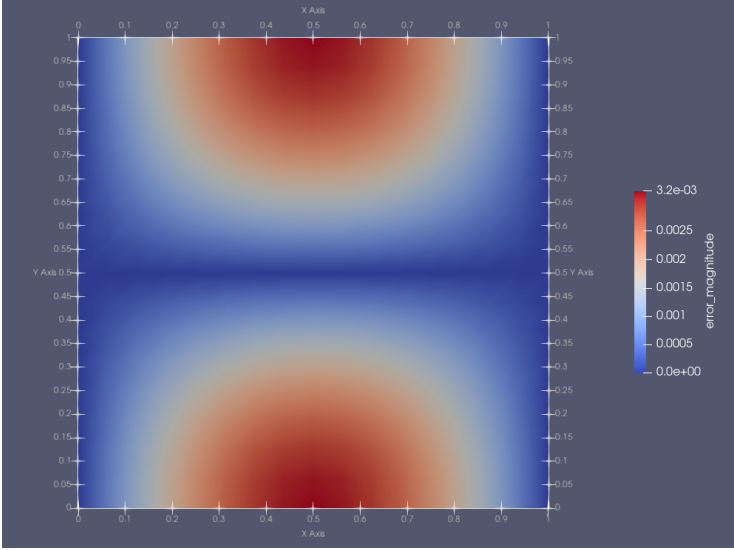


(e)  $\{h, p\} = \{256, 1\}$

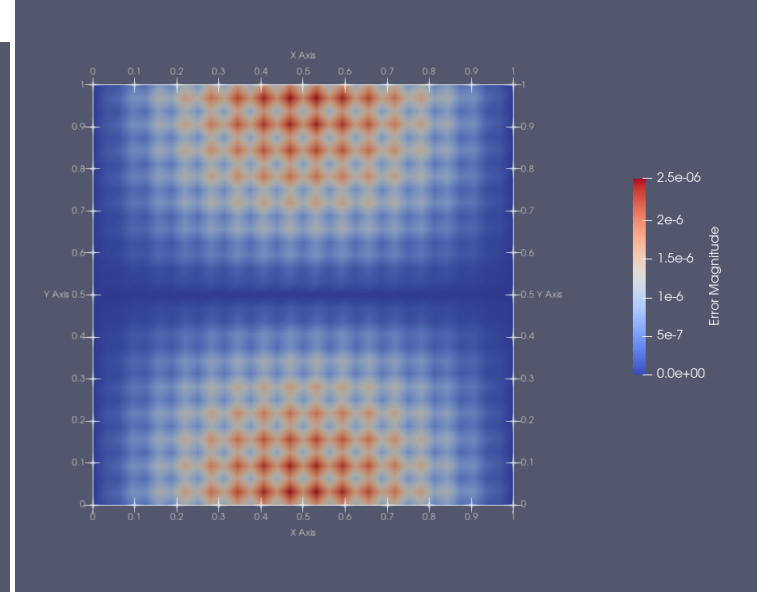


(f)  $\{h, p\} = \{512, 1\}$

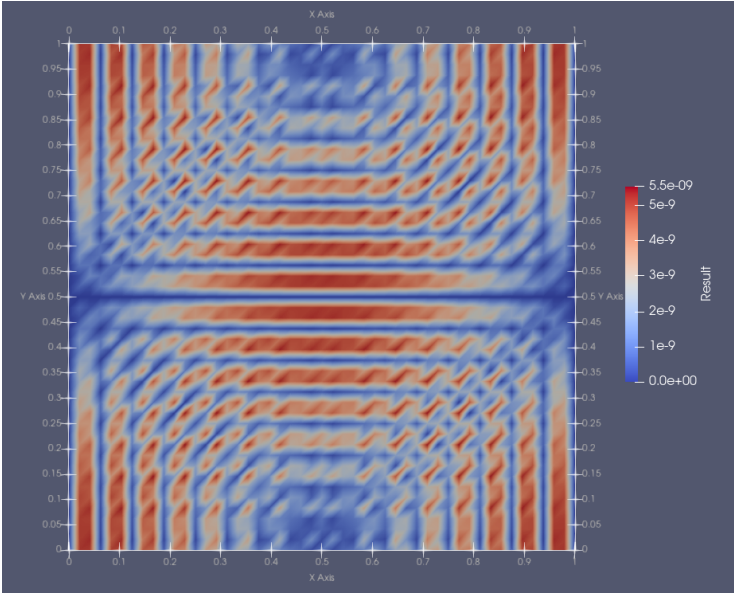
Figure 2: The same results as in Figure 1, but now each colourbar has been rescaled to show the extremities of  $|u_h - u_e|$  for the  $h$  refinement. By looking at the maxima of each scale, we observe that doubling  $h$  decreases the error by a factor of 4. Therefore, we can deduce that  $|u_h - u_e| \propto h^{-2}$  in 2D or (more generally)  $|u_h - u_e| \propto h^{-d}$  where  $d$  is the number of dimensions.



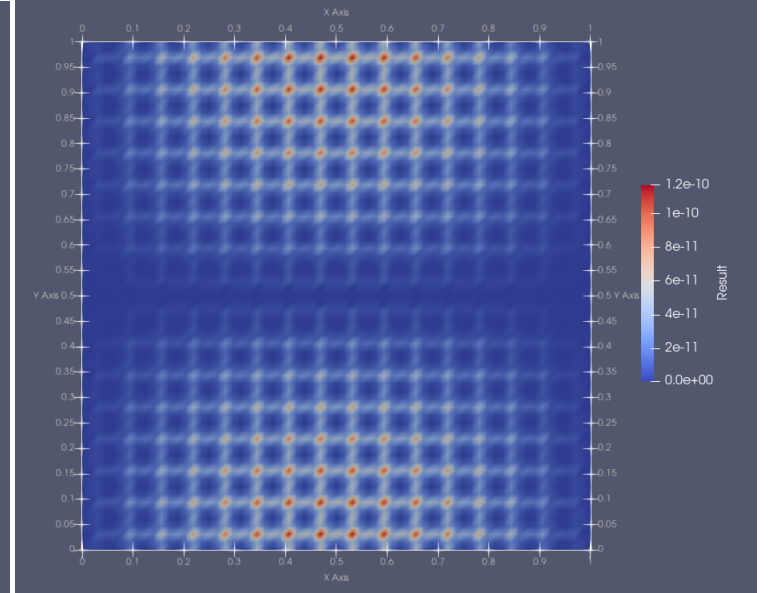
(a)  $\{h, p\} = \{16, 1\}$



(b)  $\{h, p\} = \{16, 2\}$



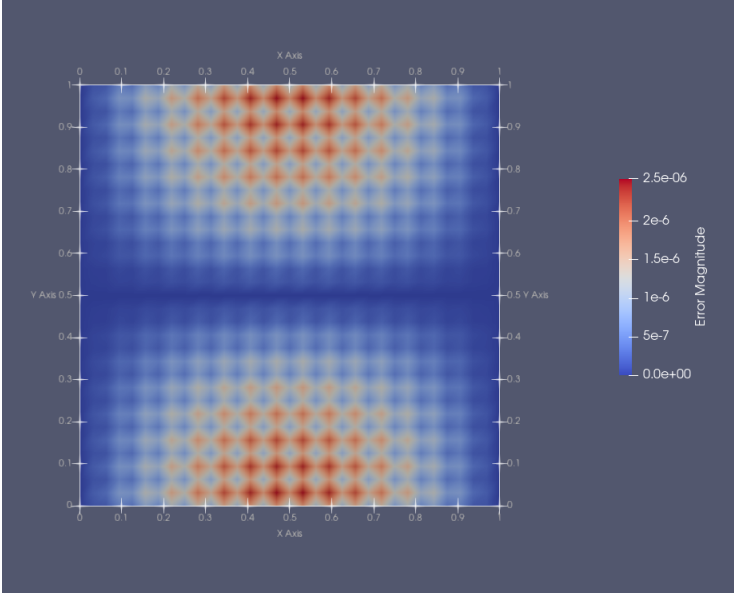
(c)  $\{h, p\} = \{16, 3\}$



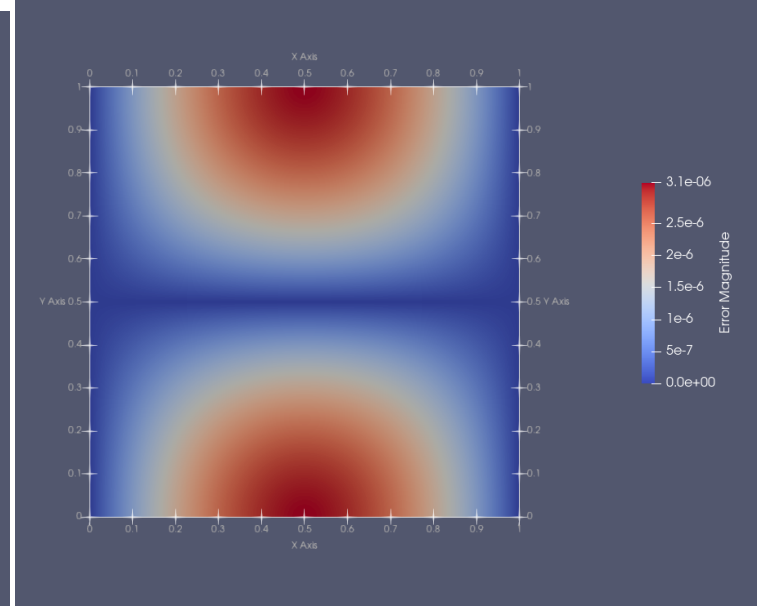
(d)  $\{h, p\} = \{16, 1\}$

Figure 3: A  $p$  refinement showing the effect of increasing the order  $p$  of the Finite Element Approximation when  $h = 16$ . Notably, increasing  $p$  greatly reduces the numerical error  $|u_h - u_e|$  but does not change the spatial resolution.

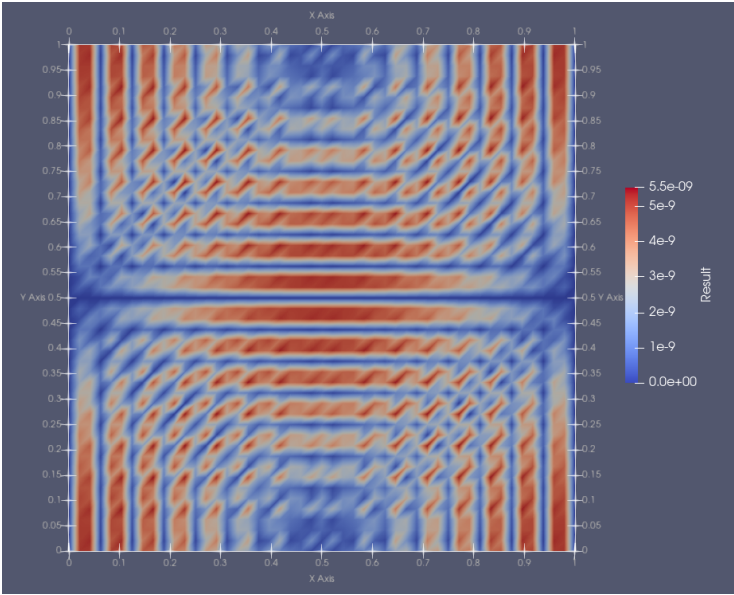




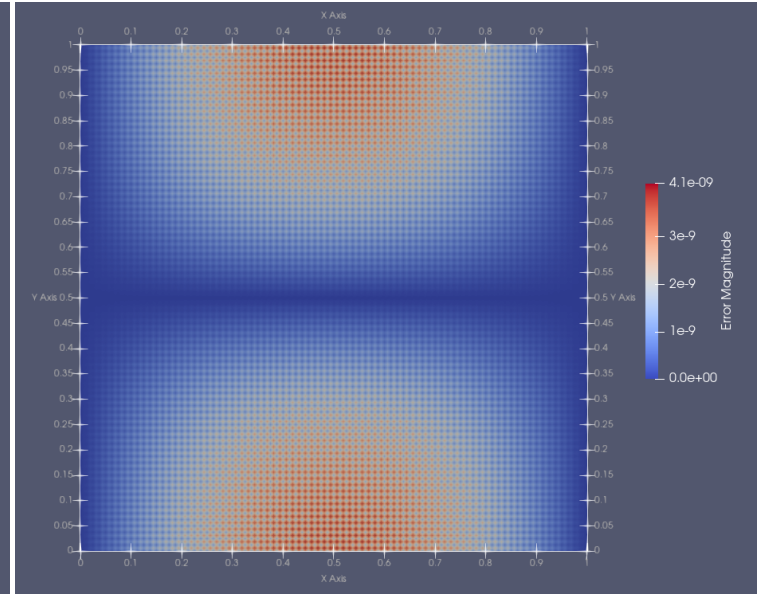
(a)  $\{h,p\} = \{16,2\}$



(b)  $\{h,p\} = \{256,1\}$

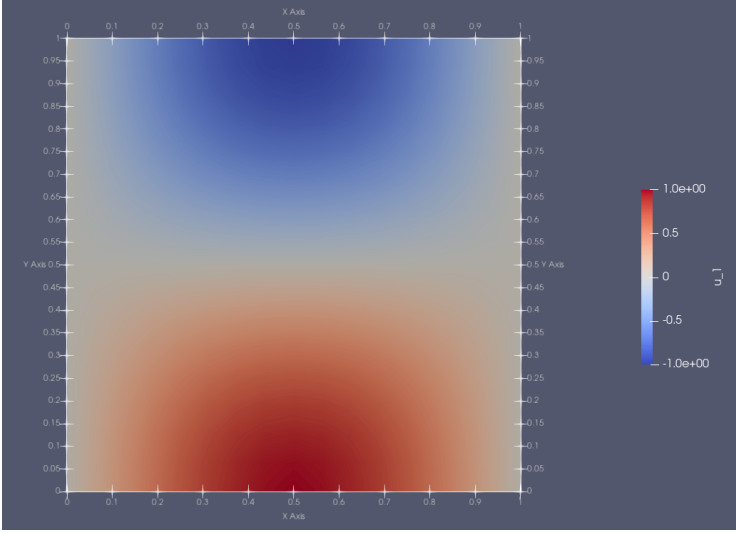


(c)  $\{h,p\} = \{16,3\}$

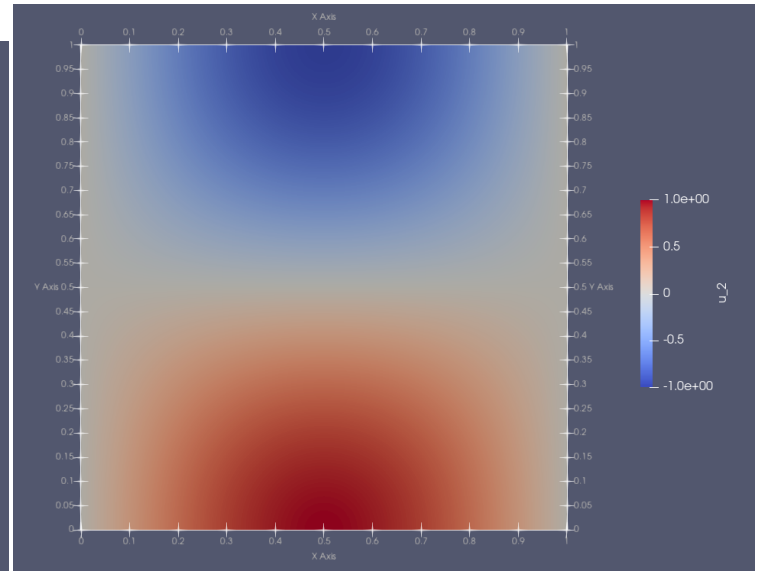


(d)  $\{h,p\} = \{80,2\}$

Figure 4: Two different  $\{h,p\}$  pairs:  $[(a),(b)]$  and  $[(c),(d)]$ . This highlights that increasing  $p$  is a very effective way of decreasing the error but does not increase the resolution of the mesh.



(a)  $\{h, p\} = \{16, 1\}$



(b)  $\{h, p\} = \{256, 1\}$

Figure 5: The numerical approximation to the Poisson equation  $u_h(x, y)$  plotted over the domain for two different values of  $h$ . Since this solution is quite simple, the low spatial resolution of (a) does not significantly affect the results that are obtained within our simulation.

Looking at these results, we can make the following deductions:

- Increasing  $h$  increases the resolution of the graph (which is most easily observed in Figure 4) and the error  $|u_h - u_e| \propto h^{-2}$  in 2D. This makes sense intuitively as the discretisation in  $x$  and  $y$  ( $\Delta x$  and  $\Delta y$ ) is given by  $\frac{1}{h}$  and  $\frac{1}{h}$  respectively for the domain  $\{0, 1\}^2$ . Therefore, we can deduce that  $|u_h - u_e| \propto \Delta q^{-1}$  where  $q$  is a generalised dimension and, in  $d$  dimensions,  $|u_h - u_e| \propto h^{-d}$  for the domain  $\{0, 1\}^d$ .
- $p$  refinement is far more effective than  $h$  refinement for minimising the error but it does not increase the spatial resolution of the mesh. Therefore, it is desirable to use a higher value of  $p$  to minimise the error but there is a trade off as you are only minimising the error at a point. Therefore, if  $h$  is too low, you do not have sufficient spatial resolution to accurately resolve the flow and you miss important flow features.
- To expand upon this, the convergence scaling which relates  $p$  and  $h$  is  $O(h^p)$  which tells you how the error decreases as  $h$  increases. This suggests that there are different scaling laws for different values of  $p$ , which makes sense intuitively as a higher order polynomial approximation samples each cell more times (it has a higher degree of freedom). This scaling causes certain  $\{h, p\}$  pairs to have a similar amount of numerical error (as observed within Figure 4).
- By comparing Figure 5 with the other figures, we observe that they share the same general profile and symmetry. Notably, the error is maximised where  $|u_h(x, y)|$  is large suggesting that an irregular mesh that places additional cells at these regions could help to resolve the flow more effectively (as an alternative to using a regular mesh and performing  $h$  or  $p$  refinement).

In conclusion,  $h$  refinement is primarily a tool for increasing the resolution of the mesh and  $p$  refinement is primarily a tool for reducing the numerical error. Utilising a mixture of the two is usually the best approach for achieving accurate and highly resolved simulations.

5) The comments have been made in my python code