

FEniCS: Three FEniCS Demonstrations related to Poisson's Equations  
Ida Ang, Edited April 4, 2023

Contents

1 Problem Definition 2

2 Formulations 2

2.1 Poisson's Equation . . . . . 2

2.2 Poisson's Equation with Periodic Boundary Condition . . . . . 4

2.3 Mixed formulation for Poisson Equation . . . . . 4

2.3.1 Fourier's Law to Weak Form . . . . . 5

2.3.2 Weak Form 2 . . . . . 6

3 FEniCS Implementation 7

3.1 Poisson's Equation . . . . . 7

3.2 Poisson's Equation with Periodic Boundary Conditions . . . . . 7

3.3 Mixed Formulation for Poisson's Equation . . . . . 8

4 Results 9

4.1 Periodic boundary condition results . . . . . 9

4.2 Comparison of single and mixed formulation . . . . . 10

# 1 Problem Definition

This document demonstrates three implementations for Poisson's equations that exists in the Programmer's reference for Dolfin (Python). The first and simplest demonstration of the Poisson Equation is given in the programmer's reference demonstration number 16, Poisson Equation Demonstration. The second demonstration is the Poisson's Equation with periodic boundary condition. The third demonstration demonstrates how to solve for the mixed formulation for Poisson's equation.

There are two keys goals to this document: 1) Fix any errors from documents that have not been updated to the last FEniCS stable version, and 2) provide more details of the theory and conversion from the strong form, or partial differential equations, to the weak form, or integral equation. This document provides the detailed steps for beginners who might have just been exposed to indicial notation. In direct notation, bolded uncapitalized notation refers to vectors, unbolded notation refers to scalar variables.

## 2 Formulations

The simplest formulation for Poisson's problem involves solving for one variable,  $u$ , which is known as a scalar field of scalar potential. The Poisson's equation shows up in many contexts, and the variable  $u$  could be the concentration of some chemical solute, as a function of position  $\mathbf{x}$ , or the temperature  $T$  in some heat conducting medium. Classically, the equation can be referred to as the divergence of the gradient of some scalar field which then returns a scalar, some source term  $f$ .

$$\nabla \cdot \nabla u = \nabla^2 u = \Delta u = f \quad \text{in } \Omega \quad (1a)$$

It can also be referred to as the Laplacian of the variable field, where several notations are listed in Eq. 1.

### 2.1 Poisson's Equation

The first setup involves Poisson's equation solved within a two dimensional square domain  $\Omega$ . The boundary consists of the Dirichlet (D) and Neumann (N) boundaries combined together:  $\partial\Omega = \Gamma_D \cup \Gamma_N$ .

$$-\nabla^2 u = f(x, y) \quad \text{in } \Omega \quad (2a)$$

$$u = 0 \quad \text{on } \Gamma_D \quad (2b)$$

$$\nabla u \cdot \mathbf{n} = \frac{\partial u}{\partial \mathbf{n}} = g \quad \text{on } \Gamma_N \quad (2c)$$

The Dirichlet boundary condition (BC) is the essential boundary condition applied within the function space, while the Neumann boundary condition is the natural boundary condition which is subsumed into the variational weak form. The gradient of scalar field gives a vector, which can be dotted with the normal vector to the boundary  $\Gamma_N$ , which is equivalent to some boundary term  $g$ . The source term  $f$  and boundary term  $g$  can be defined as follows:

$$f(x, y) = 10e^{-\frac{x^2+y^2}{0.02}} \quad (3a)$$

$$g(x) = \sin(5x) \quad (3b)$$

To see the relationship in Eq. 2c, we can rewrite the Neumann boundary condition in indicial or Einstein notation, where  $\nabla u = \frac{\partial u}{\partial x_i} \mathbf{e}_i$

$$\begin{aligned}\nabla u \cdot \mathbf{n} &= \frac{\partial u}{\partial x_i} \mathbf{e}_i \cdot n_k \mathbf{e}_k \\ &= \frac{\partial u}{\partial x_i} n_k (\mathbf{e}_i \cdot \mathbf{e}_k) \\ &= \frac{\partial u_j}{\partial x_i} n_k \delta_{ik} \\ \nabla u \cdot \mathbf{n} &= \frac{\partial u}{\partial x_i} n_i = g\end{aligned}$$

Therefore the Neumann boundary condition can be written as,

$$\frac{\partial u}{\partial x_i} n_i = g \quad (4)$$

where Eq. 2c is the same as Eq. 4. The next section goes through the conversion of the partial differential equation to the integral or weak form. First, write the laplacian in indicial notation:

$$\begin{aligned}\nabla u &= \frac{\partial u}{\partial x_i} \mathbf{e}_i \\ \nabla \cdot \nabla u &= \frac{\partial}{\partial x_j} \mathbf{e}_j \cdot \frac{\partial u}{\partial x_i} \mathbf{e}_i \\ &= \frac{\partial^2 u}{\partial x_j \partial x_i} (\mathbf{e}_j \cdot \mathbf{e}_i) \\ \nabla \cdot \nabla u &= \frac{\partial^2 u}{\partial x_i^2}\end{aligned}$$

Therefore, we can write Eq. 2

$$-\frac{\partial^2 u}{\partial x^2} = f \quad (5)$$

Multiply Eq. 5 by scalar test function,  $v$ :

$$\begin{aligned}-\frac{\partial^2 u}{\partial x^2} v &= f v \quad \text{Integrate over domain } \Omega \\ -\int_{\Omega} \frac{\partial^2 u}{\partial x^2} v dx &= \int_{\Omega} f v dx\end{aligned} \quad (6)$$

Use integration by parts,

$$(fg)' = f'g + fg' \rightarrow f'g = (fg)' - fg' \rightarrow f''g = (f'g)' - f'g' \quad (7)$$

Where we can substitute  $f'' = \frac{\partial^2 u}{\partial x_i^2}$  and  $g = v$  into Eq. 7

$$\frac{\partial^2 u}{\partial x_i^2} v = \nabla \cdot \left( \frac{\partial u}{\partial x_i} v \right) - \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i}$$

Change signs and substitute into Eq. 6

$$\begin{aligned}
& - \int_{\Omega} \nabla \cdot \left( \frac{\partial u}{\partial x} v \right) dx + \int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} dx = \int_{\Omega} f v dx \quad \text{Use divergence theorem on first term} \\
& - \int_{\partial\Omega} \left( \frac{\partial u}{\partial x} \mathbf{n} \right) v ds + \int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} dx = \int_{\Omega} f v dx \quad \text{Recognize Neumann BC Eq. 4 on first term} \\
& - \int_{\Gamma_N} g v ds + \int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} dx = \int_{\Omega} f v dx \quad u = 0 \text{ on Dirichlet boundary}
\end{aligned}$$

Therefore, the weak form in indicial and direct notation

$$\int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} dx = \int_{\Omega} f v dx + \int_{\Gamma_N} g v ds \quad (8a)$$

$$\int_{\Omega} \nabla u \cdot \nabla v dx = \int_{\Omega} f \cdot v dx + \int_{\Gamma_N} g \cdot v dx \quad (8b)$$

## 2.2 Poisson's Equation with Periodic Boundary Condition

This formulation follows as in the section above, except no Neumann Boundary condition is applied. The same square domain,  $\Omega$  is specified where the boundary,  $\partial\Omega = \Gamma_D \cup \Gamma_P$ , consists of the Dirichlet and periodic boundary conditions.

$$-\nabla \cdot (\nabla u) = f \quad \text{in } \Omega, \quad (9a)$$

$$u = 0 \quad \text{on } \Gamma_D, \quad (9b)$$

$$u(0, y) = u(1, y) \quad \text{on } \Gamma_P. \quad (9c)$$

A Dirichlet BC is applied on the top and bottom surface of the square. A periodic boundary condition means that there is some periodicity to the solutions at the boundary. In this demonstration, the solution for the potential field,  $u$ , on the right boundary is mapped to the left boundary. The implemented weak form looks as follows:

$$\int_{\Omega} \nabla u \cdot \nabla v dx = \int_{\Omega} f \cdot v dx \quad (10)$$

Note, that the source term in this case is changed (see implementation section).

## 2.3 Mixed formulation for Poisson Equation

An alternate formulation for the Poisson's equation involves introduction of a vector field,  $\mathbf{q}$ , the heat flux, through Fourier's Law. Fourier's law assumes that the heat flux is proportional to the gradient of the negative temperature

$$\begin{aligned}
\mathbf{q} &= -\kappa \nabla u \quad \kappa = -1 \\
\mathbf{q} &= \nabla u
\end{aligned} \quad (11)$$

In this particular problem, the heat conductivity,  $\kappa = -1$ , and  $u$  is the temperature field.

Consider a model for the temperature  $u$ , in a body occupying a domain  $\Omega$  subject to a heat source,  $f$ . It follows by conservation of energy that the outflow of energy over the boundary  $\partial\Omega$

must be balanced by the energy emitted by the heat source  $f$ :

$$\begin{aligned} \int_{\partial\Omega} \mathbf{q} \cdot \mathbf{n} \, ds &= - \int_{\Omega} f \, dx \quad \text{Use the Gaussian theorem} \\ \int_{\Omega} \nabla \cdot \mathbf{q} \, d\mathbf{x} &= - \int_{\Omega} f \, d\mathbf{x} \\ \nabla \cdot \mathbf{q} &= -f \quad \text{in } \Omega \end{aligned} \tag{12}$$

Note, the divergence of a vector field gives a scalar. We can thus substitute this into Poisson's equation Eq. 1.

$$\begin{aligned} -\nabla \cdot (\nabla u) &= f \quad \text{Substitute Eq. 11} \\ -\nabla \cdot \mathbf{q} &= f \end{aligned}$$

The strong form can thus be stated alongside the Neumann (Natural) and Dirichlet (Essential) boundary condition enforced on  $\Gamma_N$  and  $\Gamma_D$  respectively,

$$-\nabla \cdot \mathbf{q} = f \quad \text{in } \Omega \tag{13a}$$

$$\mathbf{q} - \nabla u = 0 \quad \text{in } \Omega \tag{13b}$$

$$u = u_o \quad \text{on } \Gamma_D \tag{13c}$$

$$\mathbf{q} \cdot \mathbf{n} = g \quad \text{on } \Gamma_N \tag{13d}$$

where we are solving for the scalar temperature field  $u$  and the vector field of heat flux  $\mathbf{q}$  in their respective function spaces. In this case, what used to be the Neumann boundary condition, the flux condition in Eq. 13d, becomes a Dirichlet or Essential boundary condition, because it must be now enforced in the function space. Meanwhile what used to be the Dirichlet boundary condition becomes the Neumann boundary condition, which must be applied in the weak form.

See the derivation for the next section. We have two partial differential equations in Eq. 13, leading to the introduction of two different test functions,  $\boldsymbol{\tau}$  and  $v$ .

### 2.3.1 Fourier's Law to Weak Form

Write Eq. 11 in indicial

$$\begin{aligned} -\frac{\partial}{\partial x_j} \mathbf{e}_j \cdot q_i \mathbf{e}_i &= f \\ -\frac{\partial q_i}{\partial x_j} \delta_{ji} &= f \\ -\frac{\partial q_i}{\partial x_i} &= f \end{aligned}$$

Next, multiply this with a scalar test function

$$\begin{aligned} -\frac{\partial q_i}{\partial x_i} v &= f v \quad \text{Integrate over domain} \\ - \int_{\Omega} \frac{\partial q_i}{\partial x_i} v \, dx &= \int_{\Omega} f v \, dx \end{aligned}$$

Change to direct notation, leaving the following variational (weak) equation.

$$- \int_{\Omega} \frac{\partial q_i}{\partial x_i} v \, dx = \int_{\Omega} f v \, dx \quad (14a)$$

$$- \int_{\Omega} (\nabla \cdot \mathbf{q}) v \, dx = \int_{\Omega} f v \, dx \quad (14b)$$

### 2.3.2 Weak Form 2

Convert Eq. 12 to indicial notation:

$$\begin{aligned} \mathbf{q} = \nabla u &\rightarrow q_i \mathbf{e}_i = \frac{\partial u}{\partial x_i} \mathbf{e}_i \quad \text{Multiply with test function } \boldsymbol{\tau} \\ q_i \mathbf{e}_i \cdot \tau_k \mathbf{e}_k &= \frac{\partial u}{\partial x_i} \mathbf{e}_i \cdot \tau_k \mathbf{e}_k \\ q_i \tau_k \delta_{ik} &= \frac{\partial u}{\partial x_i} \tau_k \delta_{ik} \\ q_i \tau_i &= \frac{\partial u}{\partial x_i} \tau_i \quad \text{Integrate over domain} \\ \int_{\Omega} q_i \tau_i \, dx &= \int_{\Omega} \frac{\partial u}{\partial x_i} \tau_i \, dx \end{aligned} \quad (15)$$

Use integration by parts on the right hand side

$$(fg)' = f'g + fg' \rightarrow f'g = (fg)' - fg' \quad (16)$$

Where we can substitute  $f' = \frac{\partial u}{\partial x_i}$  and  $g = \tau_i$  into Eq. 16

$$\frac{\partial u}{\partial x_i} \tau_i = (u\tau_i)_{,i} - u \frac{\partial \tau_i}{\partial x_i}$$

Substitute into right hand side of Eq. 16

$$\begin{aligned} \int_{\Omega} q_i \tau_i \, dx &= \int_{\Omega} \frac{\partial u}{\partial x_i} \tau_i \, dx \\ &= \int_{\Omega} (u\tau_i)_{,i} \, dx - \int_{\Omega} u \frac{\partial \tau_i}{\partial x_i} \, dx \quad \text{Use divergence theorem} \\ \int_{\Omega} q_i \tau_i \, dx &= \int_{\partial\Omega} u \tau_j n_i \, dx - \int_{\Omega} u \frac{\partial \tau_i}{\partial x_i} \, dx \end{aligned}$$

Therefore, we have the second weak forms in both indicial and direct notation below.

$$\int_{\Omega} q_i \tau_i \, dx + \int_{\Omega} u \frac{\partial \tau_i}{\partial x_i} \, dx = \int_{\partial\Omega} u \tau_j n_i \, dx \quad (17a)$$

$$\int_{\Omega} \mathbf{q} \cdot \boldsymbol{\tau} \, dx + \int_{\Omega} u (\nabla \cdot \boldsymbol{\tau}) \, dx = \int_{\Gamma} u (\boldsymbol{\tau} \cdot \mathbf{n}) \, ds \quad (17b)$$

Therefore, combining Equations 14 and 17, the weak form can be phrased as such: Find  $(\boldsymbol{\tau}, u) \in (\Sigma_g \times V)$

$$\int_{\Omega} \mathbf{q} \cdot \boldsymbol{\tau} \, dx + \int_{\Omega} (\nabla \cdot \boldsymbol{\tau}) u \, dx = \int_{\Gamma} (\boldsymbol{\tau} \cdot \mathbf{n}) u \, ds \quad \forall \mathbf{q} \in \Sigma \quad (18a)$$

$$\int_{\Omega} (\nabla \cdot \mathbf{q}) v \, dx = - \int_{\Omega} f v \, dx \quad \forall v \in V \quad (18b)$$

### 3 FEniCS Implementation

#### 3.1 Poisson's Equation

One key change from the original is the change in the mesh. Instead of using `mesh = UnitSquareMesh(32, 32)`

The mesh is set with

`mesh = RectangleMesh(Point(-0.5, -0.5), Point(0.5, 0.5), int(N), int(N))` which simply changes the origin to the center of the square. The boundaries are also specified with the following class,

```
class boundary(SubDomain):
    def inside(self, x, on_boundary):
        return x[0] < -0.5 + DOLFIN_EPS or x[0] > 0.5 - DOLFIN_EPS
```

which specifies the left hand side and the right hand side of the square for the Dirichlet boundary conditions. `DOLFIN_EPS` is simply a machine precision-like number. The boundary can be saved to a file to determine whether the boundaries are specified accurately.

#### 3.2 Poisson's Equation with Periodic Boundary Conditions

The application of the periodic boundary condition means that there is some periodicity to the solutions. In this case, the solution for the potential field on the right boundary is mapped to the left boundary. To understand this, I plotted the source term in Eq. 19, by projecting the source function into a discontinuous galerkin space.

$$f(x, y) = x \sin(5\pi y) + 1e^{-\frac{(x-0.5)^2 + (y-0.5)^2}{0.02}} \quad (19)$$

In this case, the source term has two parts, which can be superimposed over each other as demonstrated in Fig. 1. Only the first part of the function contributes to the boundary solution.

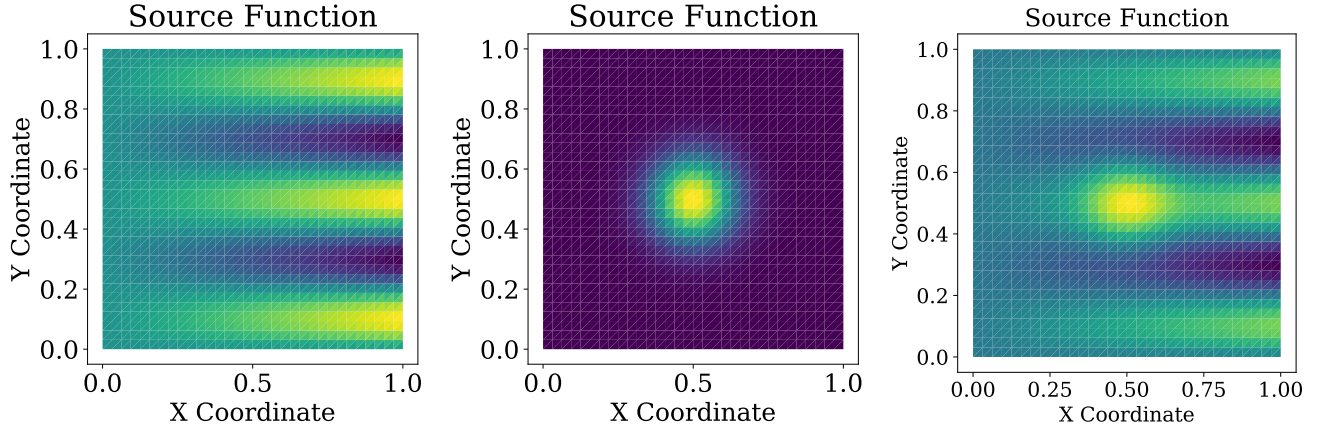


Figure 1: Visualization of the source function A) first term in Eq. 19, the B) second term in Eq. 19, and C) the full source term,  $f(x, y)$

One key change made in order to prevent a warning message from appearing is adding the last two lines to the following class to define the `value_shape`

```
class Source(UserExpression):
    def eval(self, values, x):
```

```

dx = x[0] - 0.5
dy = x[1] - 0.5
values[0] = x[0]*sin(5.0*DOLFIN_PI*x[1]) + ...
def value_shape(self):
    return ()

```

Lastly, some post-processing is included for visualization of the periodic boundary condition enforcement. This is left to give future users some idea of how to extract any field of interest.

### 3.3 Mixed Formulation for Poisson’s Equation

In this version, I kept the original mesh, where  $N$  is a parameter for the number of elements in the  $x$  and  $y$  direction

```
mesh = UnitSquareMesh(N, N)
```

The notable part of the mixed formulation is given below

```
BDM = FiniteElement("BDM", mesh.ufl_cell(), 1)
```

```
DG = FiniteElement("DG", mesh.ufl_cell(), 0)
```

```
W = FunctionSpace(mesh, BDM * DG)
```

where we find  $(\boldsymbol{\tau}, u) \in (\Sigma_g \times V)$  which is the Brezzi-Douglas-Marini (BDM) and Discontinuous Galerkin (DG) space respectively. Stability of mixed elements is a complex topic, but the order of the BDM space must be higher than the DG space.

```
(q, u) = TrialFunctions(W) (tau, v) = TestFunctions(W)
```

The boundary class wasn’t working in the last stable FEniCS release; therefore, a fix is to specify different boundaries and expressions. The boundary must also be flipped from the previous definition since the Neumann BC now becomes the Dirichlet BC.

```
class BotBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return x[1] < DOLFIN_EPS
```

```
class TopBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return x[1] > 1.0 - DOLFIN_EPS
```

It really isn’t necessary to make these classes, but this is my own preference. We know that the normal to the bottom boundary is  $\mathbf{n} = [0, -1]$  and the normal to the top boundary is  $\mathbf{n} = [0, 1]$ . The flux is meant to be enforced as a Dirichlet BC as  $\mathbf{q} \cdot \mathbf{n} = g$ . Therefore, we can write the following expressions and enforce them on the proper boundaries. An example is included below

```
g0 = Expression(("0", "sin(5*x[0])"), degree=2)
g1 = Expression(("0", "-sin(5*x[0])"), degree=2)
BotBC = DirichletBC(W.sub(0), g0, BotBoundary)
TopBC = DirichletBC(W.sub(0), g1, TopBoundary)
bc = [BotBC, TopBC]
```

Note that we are using the same expressions in Eq. 3 for this example. It is possible to further subclass the function space  $W.sub(0).sub(0)$  would indicate the  $x$ -dimension of the BDM subspace and  $W.sub(0).sub(1)$  would be the  $y$  dimension of the BDM subspace. An alternate



way to define the above snippet of code would be the following.

```
g0 = Expression(("sin(5*x[0])"), degree=2)
g1 = Expression("-sin(5*x[0])"), degree=2)
BotBC = DirichletBC(W.sub(0).sub(1), g0, BotBoundary)
TopBC = DirichletBC(W.sub(0).sub(1), g1, TopBoundary)
```

## 4 Results

### 4.1 Periodic boundary condition results

To examine whether the periodic was working properly, some post-processing is carried out where the solution profile on the right and left boundary is plotted. As expected, Fig. 2B demonstrates how the solutions match completely. Note that enforcement of periodicity, will not enforce a symmetric solution, as seen from the temperature solution as well as the plots of the profiles a small distance inwards from the right and left boundaries.

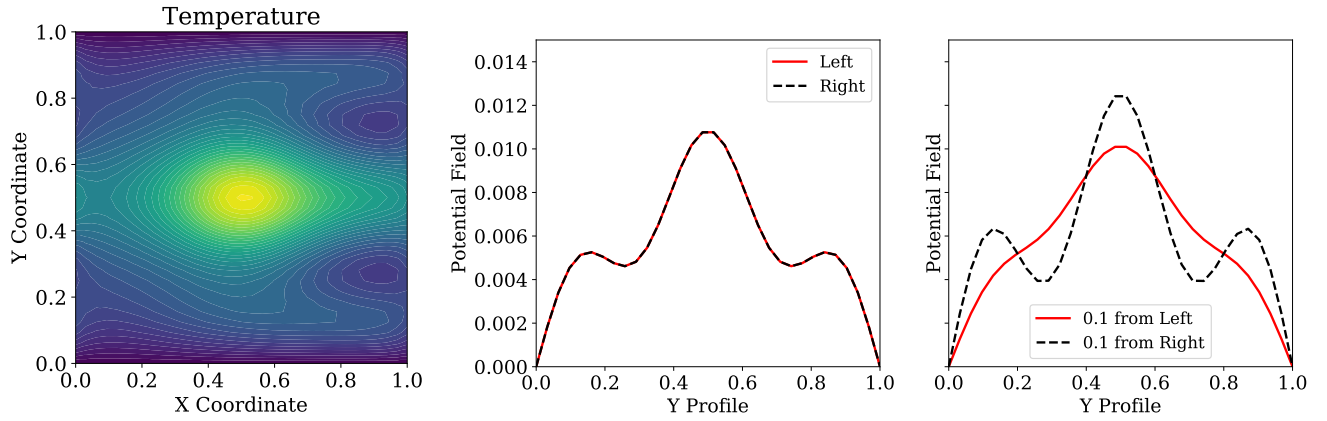


Figure 2: Temperature or potential field results. A) Potential field within the domain. B) Solution profiles on the right and left boundaries. C) Solution profiles a distance 0.1 inwards from the right and left boundaries.

## 4.2 Comparison of single and mixed formulation

Fig. 3 shows that the scalar temperature field for the results related to the first and third demonstration look quite similar, with some minor differences.

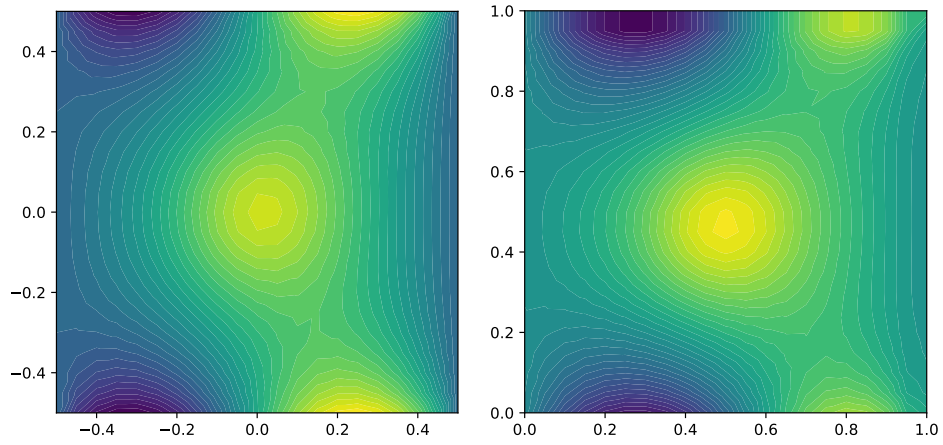


Figure 3: General Solution or temperature results from Poisson formulation and mixed Poisson Formulation respectively.

Fig. 4 displays the temperature and heat flux results from the mixed Poisson formulation

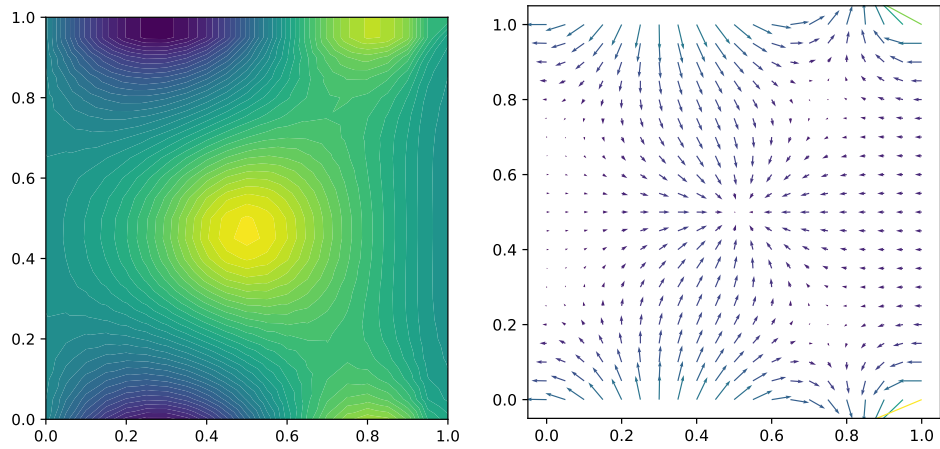


Figure 4: Mixed Poisson formulation results for a) temperature and b) heat flux.