Solving Inverse PDE by the Finite Element Method

Yahya Amad

Supervisor : Stig Larsson

CHALMERS | UNIVERSITY OF GOTHENBURG

Master thesis project in Mathematics

Name : Yahya Amad
Department : Mathematical Sciences
Institution : University of Gothenburg
Email: gusamaya@student.gu.se

Abstract

In this Master thesis project solving inverse PDE by the finite element method. An optimal control problems subjected to PDE constraint with boundary conditions is given. Construct the variational form then construct Lagrangian, which defined over whole space. Lagrangian function is function of three variables which is defined on whole space, evaluate their partial derivatives, these set of equations are the stationary point equations. Solve these stationary point equations individual, combine and into one equation by finite element method. The error, convergence rate, objective function value, error of objective function is also computed.
To calculate finite element solution used FEniCS with Python. Construct the programs in Python and run in FEniCS. Following things are discussed in the following chapters:
Chapter 1. Discussion of optimal contol problem with PDE's constraints.
Chapter 2. Discussion of the variational formulation and Lagrangian function.
Chapter 3. In this chapter discussion how to solve equations for stationary point by finite element method. To solve equations for stationary point used Python computer program language with FEniCS. FEniCS can be programmed in Python. FEniCS solves partial differential equations, this project used FEniCS to finite element equations by finite element method.
In this project a quadratic objective function subjected to linear elliptical partial differential equation with Neumann boundary condition is known, construct the variational form, Lagrangian function which is defined over whole space, taken partial derivatives of this Lagrangian function which gives set of equations are called stationary point equations, write stationary point equations as finite element solution then these equations are called finite element equations. The stationary point equations are used to find the exact solution whereas finite element equations are used to calculate finite element solution. Finally solve these finite element equations as one equation by finite element method, use programming tool FEniCS with Python.
The error analysis, convergence rate, objective function value and error of objective function are also computed. The matrix form of stationary point equations are also calculated and show that it is indefinite of saddle point form.

Contents

Chapter 1

1. **Introduction to Optimal Control**

There are various interesting problems in which cost functional has to be minimized subjected to a differential equations and partial differential equations.
Here, we used partial differential equations instead of ordinary differential equations as constraints.
There are various kinds of partial differential equations: hyperbolic partial differential equations, parabolic partial differential equations and elliptic partial differential equations but we focus of linear elliptical partial differential equations, quadratic cost functional and Neumann boundary condition.

1.1. **Boundary conditions.** The partial differential equations are combined with the conditions specified at boundary called *boundary conditions.*
We defined some boundary conditions which we used frequently in problems sections.

1.2. **Dirichlet's condition.** A homogeneous Dirichlet condition:

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

A non-homogeneous Dirichlet condition:

$$u = u_0 \quad \text{on} \quad \partial\Omega$$

1.3. **Neumann's condition.** A homogeneous Neumann condition:

$$\partial_n u = 0 \quad \text{on} \quad \partial\Omega$$

A non-homogeneous Neumann condition:

$$\partial_n u = q \quad \text{on} \quad \partial\Omega$$

1.4. **Robin's condition.** A Robin's condition is defined as:

$$\partial_n u = \alpha_0(q - u) \quad \text{on} \quad \partial\Omega$$

1.5. **Optimal boundary heating.** Consider a body which is heated or cooled under a certain temperature. The heat entering inside the body at the boundary Γ, calling heat source q, *control variable.* So the purpose is to find q such that the temperature distribution u, state variable, in Ω is best possible approximation to a required stationary temperature distribution, denoted by u_Ω or u_0 in Ω. It can be describing in mathematical way as follows:

$$\min \quad J(u,q) = \frac{1}{2}\int_\Omega |u - u_\Omega|^2 \, dx + \frac{\alpha}{2}\int_\Gamma |q|^2 \, ds$$

subjected to the PDE constraints

$$-\nabla^2 u = 0 \quad \text{in } \Omega$$
$$\partial_n u = \alpha_0(q - u) \quad \text{on } \Gamma$$

and the *point-wise control constraints*

$$q_a(x) \le q(x) \le q_b(x) \quad \text{on } \Gamma.$$

Now, we introduce the expressions which are frequently used in problems.
The surface area element is denoted by ds, unit normal on the surface boundary Γ is denoted by n, α is heat transfer coefficient from Ω to the surrounding medium. The minimized functional J is *cost functional.* The factor $1/2$ does not effect on the problem solution, it will be cancel out when taken the differential of the function. The optimal control is q and state is u. And taken negative sign with Laplacian operator, Δ, because otherwise Δ is not coercive.
Thus the cost functional is quadratic and constraints are linear elliptic partial differential equation.
Hence the whole problem is a *linear-quadratic elliptic boundary control problem.*

1.6. **Optimal heat source.** In the same way, heat source can be entered in the domain Ω. These kinds of problems heated the body Ω by microwaves or by electromagnetic induction. So in this case there is no temperature on boundary, therefore boundary temperature is zero, so that problem becomes:

$$\min \quad J(u,q) = \frac{1}{2}\int_{\Omega} |u - u_{\Omega}|^2 \, dx + \frac{\alpha}{2}\int_{\Omega} |q|^2 \, dx$$

subjected to constraints

$$-\nabla^2 u = \alpha_0 q \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \Gamma$$

$$q_a(x) \leq q(x) \leq q_b(x) \quad \text{in } \Omega$$

1.7. **Optimal non-stationary boundary control.** Suppose $\Omega \subset R^3$ denoted a potato which is roasted in oven with time $T > 0$. The temperature is denoted by $u = u(x,y)$ where $x \in \Omega,\ y \in [0,T]$. Initial temperature of the potato is $u_0 = u(x,0) = u_0(x)$ and finial temperature of potato is $u_{\Omega} = u(x,T)$. Now introduced some symbols which are used in this chapter: write $D = \Omega \times (0,T)$ and $B = \Gamma \times (0,T)$. Then the problem becomes:

$$\min \quad J(u,q) = \frac{1}{2}\int_{\Omega} |u(x,T) - u_{\Omega}(x)|^2 \, dx + \frac{\alpha}{2}\int_0^T \int_{\Gamma} |q(x,t)|^2 \, dsdt$$

subjected to PDE constraints

$$u_t - \Delta u = 0 \quad \text{in } D$$
$$\partial_n u = \alpha_0(q - u) \quad \text{on } B$$
$$u(x,0) = u_0(x) \quad \text{in } \Omega$$

and

$$q_a(x,y) \leq q(x,y) \leq q_b(x,y) \quad \text{on } B$$

this problem is *non-stationary heat equation*, which is a parabolic differential equation, where u_t represent a partial derivative of u with respect to t.

1.8. **Optimal vibrations.** Let a group of pedestrians crosses a bridge. It can be described into mathematical form as follows: suppose domain of bridge is $\Omega \subset R^2$, transversal displacement is denoted by $u(x,y)$, force density applying in vertical path is denoted $q = q(x,y)$ and transversal vibrations is denoted by $u_d = u_d(x,y)$. So that the model of optimal control problem becomes:

$$\min \quad J(u,q) = \frac{1}{2}\int_0^T \int_{\Omega} |u - u_d|^2 \, dxdt + \frac{\alpha}{2}\int_0^T \int_{\Omega} |q|^2 \, dxdt$$

subjected to constraints

$$u_{tt} - \nabla^2 u = q \quad \text{in } D$$
$$u(0) = u_0 \quad \text{in } \Omega$$
$$u_t(0) = u_1 \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } B$$

and

$$q_a(x,y) \leq q(x,y) \leq q_b(x,y) \quad \text{in} D.$$

is a *linear-quadratic hyperbolic control problem.*

Until now we discussed linear partial differential equations and we will work on linear elliptic partial differential equations but we discuss some non-linear partial differential equations: quasilinear and semi-linear equations.

1.9. **Heating with radiation boundary conditions.** If the heat radiation apply on a body then the expressions in the problem contains linear heat conduction equation with non-linear equation on boundary. In this situation PDE constraint becomes:

$$-\Delta u = 0 \quad \text{in } \Omega$$
$$\partial_n u = \alpha(q^4 - u^4) \quad \text{on } \Gamma$$

1.10. **Simplified superconductivity.** The model problem of superconductivity:

$$-\Delta u - u + u^3 = q \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \Gamma.$$

where on boundary expression of heat source is linear but in the space heat source expression is non-linear.

1.11. **Control of stationary flows.** An other model of semi-linear elliptic partial differential equation is:

$$-\frac{1}{Re}\Delta u + (u \cdot \nabla)u + \nabla q = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \Gamma$$
$$\nabla \cdot u = 0 \quad \text{in } \Omega$$

So, these are the examples of non-linear elliptic partial differential equations.
Now, we are discussing some examples of non-linear parabolic differential equations, in fact they are semi-linear.

1.12. **Problems involving semi-linear parabolic equations.** Begin with the problem involving parabolic initial-boundary value problem with temperature $u(x, y)$:

$$u_t - \Delta u = 0 \quad \text{in } D$$
$$\partial_n u = \alpha(q^4 - u^4) \quad \text{on } B$$
$$u(x, 0) = 0 \quad \text{in } \Omega.$$

In the same way, a non-stationary model of superconductivity :

$$u_t - \Delta u - u + u^3 = q \quad \text{in } D$$
$$u = 0 \quad \text{on } B$$
$$u(x, 0) = 0 \quad \text{in } \Omega.$$

1.13. **Control of non-stationary flows.** A model of *non-stationary flows of incompressible fluids* are described as follows:

$$u_t - \frac{1}{Re}\Delta u + (u \cdot \nabla)u + \nabla q = f \quad \text{in } D$$
$$\nabla \cdot u = 0 \quad \text{in } D$$
$$u = 0 \quad \text{on } B$$
$$u(x, 0) = u_0 \quad \text{in } \Omega.$$

where f volume force acts on the fluid, u_0 initial velocity and at boundary velocity is zero.

Chapter 2

2. **Variational form and Lagrangian function**

Now we discussed the variational forms and their Lagrangian's which we used in next chapter.
The variational formula is of the form :

$$A(u)(\varphi) + B(q, \psi) = 0 \text{ for all } \psi \in V$$

whereas the Lagrangian function is of the form :

$$L(u, q, \lambda) \text{ on } V \times Q \times V =: X \text{ such that}$$
$$L(u, q, \lambda) = J(u, q) - A(u)(\lambda) - B(q, \lambda)$$

Now treat the elliptic boundary value problem within the framework of Hilbert space $L^2 = L^2(\Omega)$ and derive a so-called variational formulation.

2.1. **Variational form of Poisson's equation.** An elliptic boundary value problem

$$-\nabla^2 u = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \Gamma$$

where $f \in L_2(\Omega)$. Now, to construct its variational form for all $\varphi \in C_0^\infty(\Omega)$ and integrate over Ω.

$$-\int_\Omega \varphi \nabla^2 u \, dx = \int_\Omega f\varphi \, dx$$

integrating by parts, gives

$$-\int_\Gamma \varphi \partial_n u \, ds + \int_\Omega \nabla u \cdot \nabla \varphi \, dx = \int_\Omega f\varphi \, dx$$

Since $\varphi \in C_0^\infty(\Omega)$, therefore $\varphi = 0$ on Γ, so that first term vanished.

$$\int_\Omega \nabla u \cdot \nabla \varphi \, dx = \int_\Omega f\varphi \, dx$$

So that this equation hold for all $\varphi \in C_0^\infty(\Omega)$. Since $C_0^\infty(\Omega)$ is dense in $H_0^1(\Omega)$, therefore this expression holds for all $\varphi \in H_0^1(\Omega)$.
Thus a weak formulation or variational formulation describe as below:

$$\int_\Omega \nabla u \cdot \nabla \varphi \, dx = \int_\Omega f\varphi \, dx \quad \text{for all } \varphi \in H_0^1(\Omega).$$

where the *bilinear form* $a : V \times V \to \mathbf{R}$,

$$a(u, \varphi) = \int_\Omega \nabla u \cdot \nabla \varphi \, dx$$

To write into abstract form

$$a(u, \varphi) = (f, \varphi)_\Omega \quad \text{for all } \varphi \in H_0^1(\Omega).$$

Now, we define the linear and continuous functional $F : V \to \mathbf{R}$

$$F(\varphi) = (f, \varphi)_\Omega$$

Thus, in generalized form

$$a(u, \varphi) = F(\varphi) \quad \text{for all } \varphi \in H_0^1$$

V^* is the dual space of V, the space of all linear and continuous functionals on V, where $F \in V^*$.

2.2. **Variational form of optimal control problem with Robin's condition.** Now, construct the variational form of the problem

$$-\nabla^2 u + c_0 u = f \quad \text{in } \Omega$$
$$\partial_n u + \alpha u = q \quad \text{on } \Gamma.$$

where $f \in L_2(\Omega)$ and $q \in L_2(\Gamma)$ and the coefficients, $c_0 \in L_\infty(\Omega)$ and $\alpha \in L_\infty(\Gamma)$.
So that its variational form or weak form for all $\varphi \in C^1(\bar{\Omega})$ as below
multiply the equation by φ and integrating by parts give

$$-\int_\Gamma \varphi \partial_n u \; ds + \int_\Omega \nabla u \cdot \nabla \varphi \; dx + \int_\Omega c_0 u \varphi \; dx = \int_\Omega f \varphi \; dx$$

substituting the *Robin boundary condition* gives

$$\int_\Omega \nabla u \cdot \nabla \varphi \; dx + \int_\Omega c_0 u \varphi \; dx + \int_\Gamma \alpha u \varphi \; ds = \int_\Omega f \varphi \; dx + \int_\Gamma q \varphi \; ds$$

for all $\varphi \in C^1(\bar{\Omega})$. Using the fact that $C^1(\bar{\Omega})$ is dense in $H^1(\Omega)$, this expression hold for all $\varphi \in H^1(\Omega)$.
In abstract form

$$(\nabla u, \nabla \varphi)_\Omega + c_0(u, \varphi)_\Omega + \alpha(u, \varphi)_\Gamma = (f, \varphi)_\Omega + (q, \varphi)_\Gamma \quad \text{for all } \varphi \in H^1$$

where the linear functional L is

$$L(\varphi) = \int_\Omega f \varphi \; dx + \int_\Gamma q \varphi \; ds \quad \text{for all } \varphi \in H^1$$

and bilinear form is

$$a(u, \varphi) = \int_\Omega \nabla u \cdot \nabla \varphi \; dx + \int_\Omega c_0 u \varphi \; dx + \int_\Gamma \alpha u \varphi \; ds \quad \text{for all } \varphi \in H^1$$

2.3. **Weak formulation and Lagrangian with Neumann conditions.** minimise the objective function

$$J(u, q) = \frac{1}{2}||u - u_0||^2_{\Gamma_o} + \frac{1}{2}\alpha||q||^2_{\Gamma_c}$$

where Γ_o, observational boundary and Γ_c, control boundary.$V = H^1(\Omega)$ is state space and $Q = L_2(\Gamma_c)$ is control space.

subjected to constraints

$$-\nabla^2 u + s(u) = f \text{ in } \Omega$$
$$\partial_n u = q \text{ on } \Gamma_c$$
$$\partial_n u = 0 \text{ on } \partial\Omega \setminus \Gamma_c$$

where

$$\Omega = (0,1) \times (0,1) = \{(x,y) : 0 < x < 1 \text{ and } 0 < y < 1\}, \quad \text{unit square.}$$
$$\Gamma = \{(x,y) : x = 0, y = 0, x = 1 \text{ or } y = 1\}, \quad \text{boundary of } \Omega.$$
$$\Gamma_c = \{(x,y) : y = 0\}, \quad \text{control boundary.}$$
$$\Gamma_o = \{(x,y) : y = 1\}, \quad \text{observational boundary.}$$

2.4. **Variational form and Lagrangian with homogeneous Dirichlet condition.** Minimize the optimal control problem

$$J(u, q) = \frac{1}{2}||u - u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Omega$$

where Γ is boundary, Ω is space domain and $V = H^1_0(\Omega)$, $Q = L^\infty(\Omega)$
subjected to the PDE constraint

$$-\nabla^2 u + qu = f \text{ in } \Omega$$
$$u = 0 \text{ on } \Gamma$$

since $qu = u(q)$ chosen $qu = u(q)$ as linear so that $u(q) = q$.
Construct the variational form,

$$
\begin{aligned}
-\int_\Omega \nabla^2 u \; \varphi \; dx + \int_\Omega q \; \varphi \; dx = \int_\Omega f\varphi \; dx \text{ for all } \varphi \in V \\
\int_\Gamma \varphi \; \nabla u \cdot n \; ds + \int_\Omega \nabla\varphi \cdot \nabla u \; dx + \int_\Omega q \; \varphi \; dx = \int_\Omega f\varphi \; dx \\
(\nabla\varphi, \nabla u)_\Omega + (\varphi, q)_\Omega = (\varphi, f)_\Omega \text{ for all } V \\
-(\nabla\varphi, \nabla u) - (\varphi, q) + (\varphi, f) = 0 \text{ for all } \varphi \in V
\end{aligned}
$$

Construction of Lagrangian,

$$L(u, q, \lambda) = \frac{1}{2}||u - u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Omega + (\nabla u, \nabla\lambda)_\Omega + (q, \lambda)_\Omega - (f, \lambda)_\Omega$$

2.5. **Weak form and Lagrangian with Robin's condition.** Minimize the optimal control problem

$$J(u, q) = \frac{1}{2}||u - u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Gamma$$

where Γ is boundary, Ω is space domain and $V = H^1(\Omega)$, $Q = L_2(\Gamma)$
subjected to the PDE constraint

$$
\begin{aligned}
-\nabla^2 u = 0 \text{ in } \Omega \\
\partial_n u = \nabla u \cdot n = \alpha(q - u) \text{ on } \Gamma
\end{aligned}
$$

construction of variational form

$$
\begin{aligned}
-\int_\Omega \nabla^2 u \; \varphi \; dx = 0 \text{ for all } \varphi \in V \\
-\int_\Gamma \varphi \; \nabla u \cdot n \; ds + \int_\Omega \nabla\varphi \cdot \nabla u \; dx = 0 \\
(\nabla\varphi, \nabla u)_\Omega - \alpha(\varphi, q)_\Gamma + \alpha(\varphi, u)_\Gamma = 0 \text{ for all } \varphi \in V \\
(\nabla\varphi, \nabla u)_\Omega - \alpha(\varphi, q)_\Gamma + \alpha(\varphi, u)_\Gamma = 0 \text{ for all } \varphi \in V
\end{aligned}
$$

construction of Lagrangian:

$$L(u, q, \lambda) = \frac{1}{2}||u - u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Gamma - (\nabla u, \nabla\lambda)_\Omega - \alpha(u, \lambda)_\Gamma - \alpha(q, \lambda)_\Gamma$$

2.6. **Variational form and Lagrangian with homogeneous constraints.** minimize the optimal control problem

$$J(u, q) = \frac{1}{2}||u - u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Omega$$

where Γ is boundary, Ω is space domain and $V = H^1_0(\Omega)$, $Q = L^\infty(\Omega)$
subjected to the PDE constraint

$$
\begin{aligned}
-\nabla^2 u + q = 0 \text{ in } \Omega \\
u = 0 \text{ on } \Gamma
\end{aligned}
$$

construction of variational form

$$
\begin{aligned}
-\int_\Omega \nabla^2 u \; \varphi \; dx + \int_\Omega q \; \varphi \; dx = 0 \text{ for all } \varphi \in V \\
-\int_\Gamma \varphi \; \nabla u \cdot n \; ds + \int_\Omega \nabla\varphi \cdot \nabla u \; dx + \int_\Omega q \; \varphi \; dx = 0 \\
(\nabla u, \nabla\varphi)_\Omega + (q, \varphi)_\Omega = 0 \text{ for all } \varphi \in V
\end{aligned}
$$

construction of Lagrangian:

$$L(u, q, \lambda) = \frac{1}{2}||u - u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Omega - (\nabla u, \nabla\lambda)_\Omega - (q, \lambda)_\Omega$$

2.7. **Variational form and Lagrangian with inhomogeneous constraints.** minimize the optimal control problem

$$J(u,q) = \frac{1}{2}||u-u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Gamma$$

where Γ is boundary, Ω is space domain and $V = H^1(\Omega)$, $Q = L_2(\Gamma)$
subjected to the PDE constraint

$$-\nabla^2 u = f \text{ in } \Omega$$
$$\partial_n u + u = q \text{ on } \Gamma$$

construction of variational form

$$-\int_\Omega \nabla^2 u \ \varphi \ dx = \int_\Omega f \ \varphi \ dx \quad \text{for all } \varphi \in V$$
$$-\int_\Gamma \varphi \ \nabla u \cdot n \ ds + \int_\Omega \nabla\varphi \cdot \nabla u \ dx = \int_\Omega f \ \varphi \ dx$$
$$(\nabla\varphi, \nabla u)_\Omega - \alpha(\varphi, q)_\Gamma + \alpha(\varphi, u)_\Gamma = (f, \varphi)_\Omega \quad \text{for all } \varphi \in V$$
$$(\nabla\varphi, \nabla u)_\Omega - \alpha(\varphi, q)_\Gamma + \alpha(\varphi, u)_\Gamma - (f, \varphi)_\Omega = 0 \quad \text{for all } \varphi \in V$$

construction of Lagrangian:

$$L(u,q,\lambda) = \frac{1}{2}||u-u_0||^2_\Omega + \frac{\alpha}{2}||q||^2_\Gamma - (\nabla u, \nabla\lambda)_\Omega - \alpha(u,\lambda)_\Gamma - \alpha(q,\lambda)_\Gamma - (f,\lambda)_\Omega$$

Chapter 3

3. **Finite Element Method**

In the area of partial differential equations the numerical method *finite element method* has great importance, typically elliptic and parabolic equations. The method consist on weak form of boundary value problem and approximates the solution. The symmetric positive definite elliptic problems it reduces to a finite linear system with a symmetric positive definite matrix.
Firstly we introduce the method and then solve by finite element method.
An important feature of finite element method is that the variational equations can be generated automatically by a computer program. This process is called *assembly* of the equations, which leads to a matrix and vector.

minimize the objective function, $J(u,q)$ on $V \times Q =: X$
subjected to the PDE's constraints of elliptical equation.
Construct the variational form :

$$A(u)(\varphi) + B(q,\psi) = 0 \text{ for all } \psi \in V$$

Introducing Lagrangian function:

$$L(u,q,\lambda) \text{ on } V \times Q \times V =: X \text{ such that}$$
$$L(u,q,\lambda) = J(u,q) - A(u)(\lambda) - B(q,\lambda)$$

which satisfies

$$L'_u(u,q,\lambda)(\varphi) = J'_u(u,q)(\varphi) - A'(u)(\varphi,\lambda) = 0 \quad \text{for all } \varphi \in V$$

$$L'_q(u,q,\lambda)(\chi) = J'_q(u,q)(\varphi) - B'(q)(\chi,\lambda) = 0 \text{ for all } \quad \chi \in Q$$

$$L'_\lambda(u,q,\lambda)(\psi) = -A(u)(\psi) - B(\psi,q) = 0 \quad \text{for all } \psi \in V$$

are equations for stationary point, $x := (u,q,\lambda) \in V \times Q \times V$.

Construct the strong form to calculate the exact solution (u,q,λ) and equations for stationary points are used to calculate the finite element solution (u_h,q_h,λ_h) by finite element method. Lastly compute the error, convergence rate and objective function.

3.1. **An optimal control problem.** Consider a body with domain $\Omega \subset R^2$ which want to be heated or cooled. So that enter the heat source q at boundary Γ_c, control boundary, which depends on (x,y) i.e. $q = q(x,y)$, independent of time then the goal is to find control q such that the corresponding temperature distribution $u = u(x,y)$ is the best possible approximation to a desired stationary temperature distribution u_0 on Γ_o, observational boundary. The optimal function $J(u,q)$ measures the derivation of the solution u of dirichlet values along the boundary Γ_o, observation boundary, from the prescribed function u_0.
To minimise the objective function

$$J(u,q) = \frac{1}{2}||u - u_0||^2_{\Gamma_o} + \frac{1}{2}\alpha||q||^2_{\Gamma_c}$$

where Γ_o, observational boundary and Γ_c, control boundary.
The spaces $V = H^1(\Omega)$ for state variable u and $Q = L_2(\Gamma_c)$ for control variable q.

subjected to constraints

$$
\begin{aligned}
-\nabla^2 u + s(u) &= f \text{ in } \Omega \\
\partial_n u &= q \text{ on } \Gamma_c \\
\partial_n u &= 0 \text{ on } \partial\Omega \setminus \Gamma_c
\end{aligned}
$$

where ds denotes surface element and n denotes the outward unit normal to Γ_o and Γ_c, the heat transmission coefficient from Ω to surrounding medium is represented by function α, the factor $1/2$ appears as multiple does not effect on the solution of the problem, it used just for convenience: it will be cancel out when taking the differential. The minus sign with the Laplacian operator is used to make it coercive, since Δ is not coercive operator.

$$
\begin{aligned}
\Omega &= \{(x,y) \in (0,1) \times (0,1) : 0 < x < 1 \text{ and } 0 < y < 1\}, \quad \text{unit square.} \\
\Gamma &= \{(x,y) : x = 0, y = 0, x = 1 \text{ or } y = 1\}, \quad \text{boundary of } \Omega. \\
\Gamma_c &= \{(x,y) : y = 0\}, \quad \text{control boundary.} \\
\Gamma_o &= \{(x,y) : y = 1\}, \quad \text{observational boundary.} \\
\Gamma \backslash \Gamma_o &= \{(x,y) : x = 0, y = 0, x = 1\}, \quad \text{boundary part of } \Gamma.
\end{aligned}
$$

3.2. **Construct a variational form.**

$$
\begin{aligned}
-\int_\Omega \nabla^2 u \; \varphi \; dx + \int_\Omega s(u)\varphi \; dx &= \int_\Omega f\varphi \; dx \\
-\int_\Omega \nabla \cdot \nabla u \; \varphi \; dx + \int_\Omega s(u)\varphi \; dx &= \int_\Omega f\varphi \; dx \\
-\int_\Omega \varphi \; \nabla u \cdot n \; ds + \int_\Omega \nabla u \cdot \nabla \varphi \; dx + \int_\Omega s(u)\varphi \; dx &= \int_\Omega f\varphi \; dx \\
-\int_\Omega \varphi \; \partial_n u \; ds + \int_\Omega \nabla u \cdot \nabla \varphi \; dx + \int_\Omega s(u)\varphi \; dx &= \int_\Omega f\varphi \; dx \\
-\int_{\Gamma_c} \varphi \; q \; ds + \int_\Omega \nabla u \cdot \nabla \varphi \; dx + \int_\Omega s(u)\varphi \; dx &= \int_\Omega f\varphi \; dx \\
(q,\varphi)_{\Gamma_c} - (\nabla u, \nabla\varphi)_\Omega - (s(u),\varphi)_\Omega &= -(f,\varphi)_\Omega \\
(\nabla u, \nabla\varphi)_\Omega + (s(u),\varphi)_\Omega - (f,\varphi)_\Omega - (q,\varphi)_{\Gamma_c} &= 0 \text{ for all } \varphi \in V = H^1
\end{aligned}
$$

since φ is well−defined by trace theorem: for bounded domain Ω with smooth boundary Γ, map $C^1(\Omega) \to C(\Gamma_c)$ may be extended to $H^1 \to L_2(\Gamma_c)$ with $||\varphi||_{L_2(\Gamma)} \le C||\varphi||_1$ for all $\varphi \in H^1(\Omega) = V$

$$
\begin{aligned}
A(u)(\varphi) &= (\nabla u, \nabla\varphi)_\Omega + (s(u),\varphi)_\Omega - (f,\varphi)_\Omega \text{ is semi-linear in} \varphi. \\
B(q,\varphi) &= -(q,\varphi)_{\Gamma_c} \text{ is linear.}
\end{aligned}
$$

3.3. **Construction of Lagrangian.**

$$
\begin{aligned}
L(u,q,\lambda) &= J(u,q) - A(u)(\lambda) - B(q,\lambda) \\
L(u,q,\lambda) &= \frac{1}{2}||u-u_0||^2_{\Gamma_0} + \frac{1}{2}\alpha||q||^2_{\Gamma_c} - (\nabla u,\nabla\lambda)_\Omega - (s(u),\lambda)_\Omega + (f,\lambda)_\Omega + (q,\lambda)_{\Gamma_c}
\end{aligned}
$$

where

$$
\begin{aligned}
J'_u(u,q)(\varphi) &= (\varphi, u-u_0)_{\Gamma_o} \\
J'_u(u,q)(\varphi) &= \alpha(\varphi,q)_{\Gamma_c} \\
A'(u)(\varphi,\lambda) &= (\nabla\varphi,\nabla\lambda) + (s'(u)\varphi,\lambda) \\
B'(q)(\chi,\lambda) &= -(\chi,\lambda)_{\Gamma_c} = B(\chi,\lambda)
\end{aligned}
$$

so that

$$
\begin{aligned}
L'_u(u,q,\lambda)(\varphi) &= (\varphi, u-u_0)_{\Gamma_o} - (\nabla\varphi,\nabla\lambda)_\Omega - (s'(u)\varphi,\lambda)_\Omega = 0 \text{ for all } \varphi \in H^1 \\
L'_q(u,q,\lambda)(\chi) &= \alpha(\chi,q)_{\Gamma_c} + (\chi,\lambda)_{\Gamma_c} = 0 \text{ for all } \chi \in L_2(\Gamma_c) \\
L'_\lambda(u,q,\lambda)(\psi) &= -(\nabla u,\nabla\psi)_\Omega - (s(u),\psi)_\Omega + (q,\psi)_{\Gamma_c} + (f,\psi)_\Omega = 0 \text{ for all } \psi \in H^1
\end{aligned}
$$

equations for stationary point, $x := (u, q, \lambda) \in V \times Q \times V$.

3.4. **Construct the strong form.** The advantage of the strong form is that it can be used to find exact values of the variables which are involved in the problem. These exact values are used to compare with the solution of approximate values which is helpful to compute error analysis and convergence rate.

$$L'_u(u,q,\lambda)(\varphi) = (\varphi, u-u_0)_{\Gamma_o} - (\nabla\varphi, \nabla\lambda)_\Omega - (s'(u)\varphi, \lambda)_\Omega = 0$$
$$(\varphi, u-u_0)_{\Gamma_o} - (\varphi, \nabla\lambda\cdot n)_{\partial\Omega} + (\varphi, \nabla^2\lambda)_\Omega - (\varphi, s'(u)\lambda)_\Omega = 0$$
$$(\varphi, u-u_0)_{\Gamma_o} - (\varphi, \nabla\lambda\cdot n)_{\Gamma} + (\varphi, \nabla^2\lambda - s'(u)\lambda)_\Omega = 0 \text{ for all } \psi \in H^1$$

1− for all $\varphi \in H^1_0$: $(\varphi, u-u_0)_{\Gamma_o} - (\varphi, \nabla\lambda\cdot n)_\Gamma + (\varphi, \nabla^2\lambda - s'(u)\lambda)_\Omega = 0$
Since $\varphi \in H^1_0$ therefore $\varphi = 0$ on Γ_o, Γ
$\nabla^2\lambda - s^{'}(u)\lambda = 0$ in Ω
2− for all $\varphi \in H^1$: $(\varphi, u-u_0-\nabla\lambda\cdot n)_{\Gamma_o} - (\varphi, \nabla\lambda\cdot n)_{\Gamma\setminus\Gamma_o} = 0$
this implies that
$-\nabla\lambda\cdot n = -\partial_n\lambda = 0$ on $\Gamma \setminus \Gamma_o$
$u - u_0 - \nabla\lambda\cdot n = 0$ on Γ_0
$u - u_0 = \partial_n\lambda$ on Γ_0
The strong form of λ equation

$$\boxed{\begin{array}{c} -\nabla^2\lambda + \lambda = 0 \text{ in } \Omega \\ \partial_n\lambda = 0 \text{ on } \partial\Omega\setminus\Gamma_o \\ u-u_0 = \partial_n\lambda \text{ on } \Gamma_o \end{array}}$$

$$L'_q(u,q,\lambda)(\chi) = \alpha(\chi, q)_{\Gamma_c} + (\chi, \lambda)_{\Gamma_c} = (\chi, \alpha q + \lambda)_{\Gamma_c} = 0 \text{ for all } \chi \in L^2(\Gamma_c)$$

The strong form of q equation

$$\boxed{\alpha q + \lambda = 0 \text{ on } \Gamma_c}$$

$$L'_\lambda(u,q,\lambda)(\psi) = -(\nabla u, \nabla\psi)_\Omega - (s(u), \psi)_\Omega + (q, \psi)_{\Gamma_c} + (f, \varphi) = 0$$
$$(\nabla^2 u, \psi)_\Omega - (\nabla u\cdot n, \psi)_\Gamma - (s(u), \psi)_\Omega + (q, \psi)_{\Gamma_c} + (f, \phi) = 0$$
$$(\nabla^2 u, \psi)_\Omega - (\nabla u\cdot n, \psi)_{\Gamma\setminus\Gamma_c} - (\nabla u\cdot n, \psi)_{\Gamma_c} - (s(u), \psi)_\Omega + (q, \psi)_{\Gamma_c} + (f, \phi) = 0$$
$$(-\nabla u\cdot n, \psi)_{\Gamma\setminus\Gamma_c} + (\nabla^2 u - s(u) + f, \psi)_\Omega + (-\nabla u\cdot n + q, \psi)_{\Gamma_c} = 0 \text{ for all } \psi \in H^1$$

1− for all $\psi \in H^1_0$: $(-\nabla u\cdot n, \psi)_{\Gamma\setminus\Gamma_c} + (\nabla^2 u - s(u) + f, \psi)_\Omega + (-\nabla u\cdot n + q, \psi)_{\Gamma_c} = 0$
$\nabla^2 u - s(u) + f = 0$ in Ω
$-\nabla^2 u + s(u) = f$ in Ω
2− for all $\psi \in H^1$: $(-\nabla u\cdot n + q, \psi)_{\Gamma_c} = 0$
3− $\psi = 0$ on Γ_c : $(-\nabla u\cdot n, \psi)_{\partial\Omega\setminus\Gamma_c} = 0$
$-\partial_n u = -\nabla u = 0$ on $\partial\Omega \setminus \Gamma_c$
4− for all $\psi \in H^1$: $(-\nabla u\cdot n + q, \psi)_{\Gamma_c} = 0$
$-\nabla u\cdot n + q = 0$ on Γ_c
$\partial_n u = \nabla u\cdot n = q$ on Γ_c
The strong form of u equation

$$\boxed{\begin{array}{c} -\nabla^2 u + s(u) = f \text{ in } \Omega \\ \partial_n u = q \text{ on } \Gamma_c \\ \partial_n u = 0 \text{ on } \partial\Omega\setminus\Gamma_c \end{array}}$$

describe the stationary point equations as finite element equations.

$$
\begin{aligned}
L'_u(u_h,q_h,\lambda_h)(\varphi) &= \rho_h^*(u_h,q_h,\lambda_h)(\varphi_h) = (\varphi_h,u_h-u_0)_{\Gamma_o} - (\nabla\varphi_h,\nabla\lambda_h)_\Omega - (s'(u_h)\varphi_h,\lambda_h)_\Omega = 0\\
&\forall\ \varphi_h \in H^1\\
L'_q(u_h,q_h,\lambda_h)(\chi) &= \rho_h^q(u_h,q_h,\lambda_h)(\chi_h) = \alpha(\chi_h,q_h)_{\Gamma_c} + (\chi_h,\lambda_h)_{\Gamma_c} = 0\\
&\forall \chi_h \in L_2(\Gamma_c)\\
L'_\lambda(u_h,q_h,\lambda_h)(\psi_h) &= \rho_h(u_h,q_h,\lambda_h)(\psi_h) = -(\nabla u_h,\nabla\psi_h)_\Omega - (s(u_h),\psi_h)_\Omega + (q_h,\psi_h)_{\Gamma_c} + (f,\psi_h)\\
&\forall \psi_h \in H^1
\end{aligned}
$$

3.5. **System is indefinite of saddle point form.** Consider $u_h = \sum_{i=1}^n u_i\Phi_i(x)$, $\lambda_h = \sum_{i=1}^n \lambda_i\Phi_i(x)$, $q_h = \sum_{i=1}^n q_i\Phi_i(x)$
since Ω is unit square. Used $s(u)$ as linear i.e. $s(u) = u$ therefore $s'(u) = 1$.
using $\Phi_i(x)$ as test function defined as

$$
\Phi_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}
$$

since

$$
\begin{aligned}
L'_u(u,q,\lambda)(\varphi) = (\varphi,u-u_0)_{\Gamma_o} - (\nabla\varphi,\nabla\lambda)_\Omega - (s'(u)\varphi,\lambda)_\Omega &= 0\\
(\varphi,u)_{\Gamma_o} - (\varphi,u_0)_{\Gamma_o} - (\nabla\varphi,\nabla\lambda)_\Omega - (\varphi,\lambda)_\Omega &= 0\\
(\Phi_j,\sum_{i=1}^n u_i\Phi_i(x))_{\Gamma_o} - (\Phi_j,\sum_{i=1}^n u_{0i}\Phi_i(x))_{\Gamma_o} - (\nabla\Phi_j,\nabla\sum_{i=1}^n \lambda_i\Phi_i(x))_\Omega - (\Phi_j,\sum_{i=1}^n \lambda_i\Phi_i(x))_\Omega &= 0\\
(\Phi_j,\sum_{i=1}^n (u_i-u_{0i})\Phi_i(x))_{\Gamma_o} - \sum_{i=1}^n \lambda_i(\nabla\Phi_j,\nabla\Phi_i(x))_\Omega - \sum_{i=1}^n \lambda_i(\Phi_j,\Phi_i(x))_\Omega &= 0\\
\sum_{i=1}^n (u_i-u_{0i})(\Phi_j,\Phi_i(x))_{\Gamma_o} - \sum_{i=1}^n \lambda_i(\nabla\Phi_j,\nabla\Phi_i(x))_\Omega - \sum_{i=1}^n \lambda_i(\Phi_j,\Phi_i(x))_\Omega &= 0
\end{aligned}
$$

λ equation into the matrix form

$$
[u_1-u_{10}u_2-u_{20}\cdots u_n-u_{n0}]\cdot\begin{bmatrix}(\Phi_j,\Phi_1)_{\Gamma_o}\\(\Phi_j,\Phi_2)_{\Gamma_o}\\ \cdot\\ \cdot\\ \cdot\\(\Phi_j,\Phi_n)_{\Gamma_o}\end{bmatrix} - [\lambda_1\lambda_2\cdots\lambda_n]\cdot\begin{bmatrix}(\nabla\Phi_j,\nabla\Phi_1)_\Omega\\(\nabla\Phi_j,\nabla\Phi_2)_\Omega\\ \cdot\\ \cdot\\ \cdot\\(\nabla\Phi_j,\nabla\Phi_n)_\Omega\end{bmatrix}
- [\lambda_1\lambda_2\cdots\lambda_n]\cdot\begin{bmatrix}(\Phi_j,\Phi_1)_\Omega\\(\Phi_j,\Phi_2)_\Omega\\ \cdot\\ \cdot\\ \cdot\\(\Phi_j,\Phi_n)_\Omega\end{bmatrix} = 0
$$

or

$$
\begin{aligned}
(\bar{u}-\bar{u_0})^T A_{\Gamma_o} - \bar{\lambda}^T B_\Omega - \bar{\lambda}^T A_\Omega &= 0\\
A_{\Gamma_o}^T(\bar{u}-\bar{u_0}) - B_\Omega^T\bar{\lambda} - A_\Omega^T\bar{\lambda} &= 0 \qquad \text{........} \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
L'_q(u,q,\lambda)(\chi) = \alpha(\chi,q)_{\Gamma_c} + (\chi,\lambda)_{\Gamma_c} = (\chi,\alpha q+\lambda)_{\Gamma_c} = 0 \\
(\chi,\alpha q+\lambda)_{\Gamma_c} = 0 \text{ for all } \chi \in L_2(\Gamma_c) \\
(\Phi_j,\alpha\sum_{i=1}^{n} q_i\Phi_i(x))_{\Gamma_c} + (\Phi_j,\sum_{i=1}^{n}\lambda_i\Phi_i(x)) \\
\sum_{i=1}^{n}(\Phi_j,(\alpha q_i+\lambda_i)\Phi_i(x))_{\Gamma_c} = 0 \\
\sum_{i=1}^{n}(\alpha q_i+\lambda_i)(\Phi_j,\Phi_i(x))_{\Gamma_c} = 0
\end{aligned}
$$

q equation into the matrix form

$$
[\alpha q_1+\lambda_1 \ \ \alpha q_2+\lambda_2 \ \ \cdots \alpha q_n+\lambda_n]\cdot\begin{bmatrix}(\Phi_j,\Phi_1)_{\Gamma_c}\\(\Phi_j,\Phi_2)_{\Gamma_c}\\ \cdot\\ \cdot\\ \cdot\\(\Phi_j,\Phi_n)_{\Gamma_c}\end{bmatrix} = 0
$$

or

$$
\begin{aligned}
-\alpha\bar{q}^T A_{\Gamma_c} = \bar{\lambda}^T A_{\Gamma_c} \\
\alpha A^T_{\Gamma_c}\bar{q} + A^T_{\Gamma_c}\bar{\lambda} = 0 \qquad \ldots\ldots\ldots \qquad (2)
\end{aligned}
$$

$$
\begin{aligned}
L'_\lambda(u,q,\lambda)(\psi) = -(\nabla u,\nabla\psi)_\Omega - (u,\psi)_\Omega + (q,\psi)_{\Gamma_c} + (f,\psi)_\Omega = 0 \text{ for all } \psi\in H^1 \\
-(\nabla u,\nabla\psi)_\Omega - (u,\psi)_\Omega + (q,\psi)_{\Gamma_c} + (f,\psi)_\Omega = 0 \text{ for all } \psi\in H^1 \\
-(\nabla\sum_{i=1}^{n}u_i\Phi_i(x),\nabla\Phi_j)_\Omega - (u,\Phi_j)_\Omega + (\sum_{i=1}^{n}q_i\Phi_i(x),\Phi_j)_{\Gamma_c} + (\sum_{i=1}^{n}f_i\Phi_i(x),\Phi_i)_\Omega = 0 \\
-\sum_{i=1}^{n}u_i(\nabla\Phi_i,\nabla\Phi_j)_\Omega - \sum_{i=1}^{n}u_i(\Phi_i,\Phi_j)_\Omega + \sum_{i=1}^{n}q_i(\Phi_i,\Phi_j)_{\Gamma_c} + \sum_{i=1}^{n}f_i(\Phi_i,\Phi_j)_\Omega = 0 \\
-\sum_{i=1}^{n}u_i a(\Phi_i,\Phi_j)_\Omega - \sum_{i=1}^{n}u_i(\Phi_i,\Phi_j)_\Omega + \sum_{i=1}^{n}q_i(\Phi_i,\Phi_j)_{\Gamma_c} + \sum_{i=1}^{n}f_i(\Phi_i,\Phi_j)_\Omega = 0
\end{aligned}
$$

u equation into the matrix form

$$
-[u_1u_2\cdots u_n]\cdot\begin{bmatrix}a(\Phi_j,\Phi_1)_\Omega\\a(\Phi_j,\Phi_2)_\Omega\\ \cdot\\ \cdot\\ \cdot\\a(\Phi_j,\Phi_n)_\Omega\end{bmatrix} + [-u_1+f_1 \quad -u_2+f_2\cdots -u_n+f_n]\cdot\begin{bmatrix}(\Phi_j,\Phi_1)_\Omega\\(\Phi_j,\Phi_2)_\Omega\\ \cdot\\ \cdot\\ \cdot\\(\Phi_j,\Phi_n)_\Omega\end{bmatrix}
$$

$$
+[q_1 \quad q_2\cdots q_n]\cdot\begin{bmatrix}(\Phi_j,\Phi_1)_{\Gamma_c}\\(\Phi_j,\Phi_2)_{\Gamma_c}\\ \cdot\\ \cdot\\ \cdot\\(\Phi_j,\Phi_n)_{\Gamma_c}\end{bmatrix} = 0
$$

$$
\begin{aligned}
-\bar{u}^T B_\Omega + (-\bar{u}+\bar{f})^T A_\Omega + \bar{q}^T A_{\Gamma_c} = 0 \\
-(B_\Omega + A_\Omega)\bar{u}^T + A_{\Gamma_c}\bar{q}^T = -f^T A_\Omega \qquad \ldots\ldots\ldots \qquad (3)
\end{aligned}
$$

write the following equations:

$$
\begin{aligned}
A^T_{\Gamma_o}\bar{u} - (B^T_\Omega + A^T_\Omega)\bar{\lambda} = A^T_{\Gamma_o}u_0 \\
\alpha A^T_{\Gamma_c}\bar{q} + A^T_{\Gamma_c}\bar{\lambda} = 0 \\
-(B_\Omega + A_\Omega)\bar{u} + A_{\Gamma_c}\bar{q} = -A_\Omega f
\end{aligned}
$$

into matrix form as follows:

$$\left[\begin{array}{c|c|c} A_{\Gamma_o}^T & 0 & -(B_\Omega^T + A_\Omega^T) \\ \hline 0 & \alpha A_{\Gamma_c}^T & A_{\Gamma_c}^T \\ \hline -(B_\Omega + A_\Omega) & A_{\Gamma_c} & 0 \end{array}\right] \begin{bmatrix} u \\ q \\ \lambda \end{bmatrix} = \begin{bmatrix} A_{\Gamma_o}^T u_0 \\ 0 \\ -A_\Omega f \end{bmatrix}$$

the block form of the matrix is :

$$\begin{bmatrix} A_e & B_e^T \\ B_e & 0 \end{bmatrix}$$

where

$$A_e = \begin{bmatrix} A_{\Gamma_o}^T & 0 \\ 0 & \alpha A_{\Gamma_c}^T \end{bmatrix}, B_e = \begin{bmatrix} -(B_\Omega + A_\Omega) & A_{\Gamma_c} \end{bmatrix},$$

$$B_e^T = \begin{bmatrix} -(B_\Omega^T + A_\Omega^T) \\ A_{\Gamma_c}^T \end{bmatrix}, b_e = \begin{bmatrix} A_{\Gamma_o}^T u_0 \\ 0 \\ -f A_\Omega \end{bmatrix}$$

The block form of the matrix has eigenvalues $\frac{A_e}{2} \pm \sqrt{\frac{A_e{}^2}{4} + B_e B_e^T}$, one positive, one negative, therefore it is a saddle point.

λ equation

$$L'_u(u, q, \lambda)(\varphi) = (\varphi, u - u_0)_{\Gamma_o} - (\nabla\varphi, \nabla\lambda)_\Omega - (\varphi, \lambda)_\Omega = 0 \text{ for all } \varphi \in H^1$$

q equation

$$L'_q(u, q, \lambda)(\chi) = \alpha(\chi, q)_{\Gamma_c} + (\chi, \lambda)_{\Gamma_c} = 0 \text{ for all } \chi \in L_2(\Gamma_c)$$

u equation

$$L'_\lambda(u, q, \lambda)(\psi) = -(\nabla u, \nabla\psi)_\Omega - (u, \psi)_\Omega + (q, \psi)_{\Gamma_c} + (f, \psi) = 0 \text{ for all } \psi \in H^1$$

3.6. **Finite element calculation.** In this section choose an objective function subjected to PDE constraint with boundary condition. Construct the variational form. Then the Lagrangian function. Taken partial derivatives of the Lagrangian function. These set of equations are called stationary point equations. Write these stationary point equations in term of approximate solution. Then these equations are called finite element equations. Now solve these finite element equations individual, combine and one equation by finite element method.

3.7. **Solve stationary point equations individual by FEM.** In this section solve the finite element equations individually, choose some variable as variable function or constant function and solve other variables. To solve stationary point equations by finite element method used the objective function subjected to PDE's constraints. To solve finite element equations individual uesd this case as an test example.

Minimise the objective function

$$J(u, q) = \frac{1}{2}||u - u_0||^2_{\Gamma_o} + \frac{1}{2}\alpha||q||^2_{\Gamma_c}$$

where Γ_o, observational boundary and Γ_c, control boundary.

subjected to constraints

$$-\nabla^2 u + s(u) = f \text{ in } \Omega$$
$$\partial_n u = q \text{ on } \Gamma_c$$
$$\partial_n u = 0 \text{ on } \partial\Omega \setminus \Gamma_c$$

Choose q, f and solve for u in strong form for exact solution. Chosen $s(u)$ as linear function of u such that $s(u) = u$ and $s'(u) = 1$. Now our task is to construct an exact solution to be used as an test example when $q = 1.0$ on Γ_c and $f = \frac{y^2}{2} - y + \frac{1}{2}$. Chosen such value of q find u which satisfies the boundary conditions and satisfies the equation in strong form this gives the value of f when we substitute u in the given strong form. So that we obtain following u function.

$$u = \frac{y^2}{2} - y + \frac{3}{2}$$

Compute their values corresponding to various meshes. Continuously computing the values of u corresponding to different step size. The maximum and minimum u values when mesh, $h = \frac{1}{4}$, is $u = 1.5$ and $u = 1.0$ respectively. So that the maximum and minimum values of u can be obtained when $y = 0$ and $y = 1$, from the exact solution, which are the boundary parts of Γ_c and Γ_o respectively. Thus the maximum value of u lies on the boundary Γ_c where $y = 0$ and the minimum value of u lies on the boundary Γ_o where $y = 1$.
For arbitrary q the exact solution of u is, whose first term vanished when $q = 1$:

$$u = (q-1)\frac{\cosh(y-1)}{\sinh(1)} + \frac{y^2}{2} - y + \frac{3}{2}$$

Choose q, f and solve for u_h in variational form for finite element solution. The method which used in place of an approximate solution is *finite element method* which is computed in python with FEniCS. FEniCS is working as operating system like for windows used command prompt and Linux used terminal prompt. Now, chosen $q = 1.0$ and $f = \frac{y^2}{2} - y + \frac{1}{2}$ compute u_h - value by finite element method corresponding to various meshes. Thus the maximum and minimum u_h values when mesh is chosen, $h = \frac{1}{4}$, is $u_h = 1.5025067483$ and $u_h = 0.997524820386$ respectively.
To solve the variational equation used *assemble* which is integrated whole equation and it include matrix and vector. Thus it is one of the advantage that we interpolate whole equation as it is and then it automatically loaded vector and matrix to solve the equation by finite element method in python. Formulas in L^2 which are frequently used to calculate error and convergence rate corresponding to different values of step size h. To computing the error in python used L^2 norm where we used term *assemble* which is integrated over whole space.

$$||u - u_h||_{L^2} = (\int |u - u_h|^2 \ dx)^{1/2}$$

$$r_u = \frac{\ln(E_i/E_{i-1})}{\ln(h_i/h_{i-1})}$$

It is an error and convergence rate corresponding to different meshes and the exact solution is interpolated onto the space of degree three.The approximate solution is interpolated onto the space of degree one whereas the exact solution is interpolated onto the space of degree three. This gives better analysis.
It can be observed that error is decreasing with an approximate factor $\frac{1}{4}$ as step size h is decreasing.

TABLE 1. Error Analysis

h	$\vert\vert u - u_h\vert\vert_{L^2}$
1/4	0.00602560985088000
1/8	0.00151851498406000
1/16	0.00038058129942700
1/32	0.00009521763865940
1/64	0.00002380988317370
1/128	0.00000595339400009

TABLE 2. Convergence rate

h	$r_{u_{L^2}}$
1/4	1.98845
1/8	1.99638
1/16	1.99890
1/32	1.99967
1/64	1.99978

REMARK . For solution in finite element method used a subscription h with solution value for instant u_h. It denotes the solution of the *finite element method.*

Program syntax for u in Python.
Define the space V.

```
V = FunctionSpace(mesh, "Lagrange", 1)
```

Specify the boundary parts Γ_o, Γ_c and $\Gamma\backslash\Gamma_o$

```
boundary-parts = MeshFunction("uint", mesh, 1)
tol = 1E-14
class LowerBoundary(SubDomain):
def inside(self, x, on_boundary):
return on_boundary and abs(x[1]) < tol
Gamma_c = LowerBoundary()
Gamma_c.mark(boundary_parts, 0)
```

The control boundary Γ_c is marked by 0, the observational boundary Γ_o is marked by 1.

```
class UpperBoundary(SubDomain):
    def inside(self, x, on-boundary):
        return on_boundary and abs(x[1]-1) < tol
Gamma_o = UpperBoundary()
Gamma_o.mark(boundary_parts, 1)
```

The rest of the boundary, $\Gamma\backslash\Gamma_o$ is defined as:

```
class RestBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and abs(x[0]) < tol or abs(x[1]-1) < tol or abs(x[0]-1) < tol
Gamma_{t_c} = RestBoundary()
Gamma_{t_c}.mark(boundary_parts, 2)
```

Define state variable as trial function, control variable as constant, test function, exact solution of state variable , function f.

```
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("(x[1]*x[1]/2)-x[1]+0.5")
q = Constant("1.0")
Ve = FunctionSpace(mesh, "Lagrange" , degree=3)
u_a = Expression("-x[1]+(x[1]*x[1]/2)+1.5")
u_ext = interpolate(u_a,Ve)
```

Define variational equation, variable term define on one side and constant part is define on the other side.

```
a = inner(grad(u), grad(v))*dx + inner(u,v)*dx
L = f*v*dx + q*v*ds(0)
```

This syntax used to integrate variational equation and print syntax used to show the variational form into matrix form.

```
A = assemble(a, exterior_facet_domains=boundary_parts)
b = assemble(L, exterior_facet_domains=boundary_parts)

print'A : ', A.array()
print'b : ', b.array()
```

To define solution variable.

```
u = Function(V)
U = u.vector()
solve(A, U, b)
```

Define error norm in space L^2.

```
error_2 = inner(u - u_ext,u - u_ext)*dx
E = sqrt(assemble(error_2))
```

Choose $u - u_0$ and solve for λ in strong form for exact solution.
Now task is to solve the variational equation for λ keeping $u - u_0$ known. Previously solved for u keeping q known. For exact solution, chosen $u - u_0 = 1.0$ on Γ_o, so that the strong form give the $\lambda-$value by using boundary conditions. The λ-value is function of exponential and there is multiple +1 thus the whole expression gives positive value. The expression can be expressed in term of *hyperbolic cosine* and *hyperbolic sine* functions.

$$\lambda = \frac{1}{e - e^{-1}}(e^y + e^{-y}) = \frac{\cosh(y)}{\sinh(1)}$$

Compute their values corresponding to various meshes. The minimum and maximum exact λ values when $h = \frac{1}{4}$ is $\lambda = 0.850918128239$ and $\lambda = 1.3130352855$ respectively.
Chosen $\alpha = 1$ so that $\lambda = -\alpha q = -q$ on $\Gamma_c = \{(x, y) \in (0, 1) \times (0, 1) : y = 0\}$
So that $\lambda = \frac{cosh(0)}{sinh(1)} = 0.8509181282393216$ on Γ_c which is minimum value of λ and $\lambda = \frac{cosh(1)}{sinh(1)} = 1.3130352854993312$ on Γ_o which is maximum value of λ over space Ω.
Thus the maximum value of λ is occurred on the boundary Γ_o where $y = 1$ and the minimum value of λ is occurred on the boundary Γ_c where $y = 0$.
Choose $u - u_0$ and solve for λ_h in variational form finite element solution.
Now the aim is to calculate λ_h-value corresponding to various meshes by *finite element method*. The method which used as an approximate solution is *finite element method* which is computed in python. Now, chosen $u - u_0 = 1.0$ and compute λ_h - value by finite element method corresponding to different meshes. When chosen mesh, $h = \frac{1}{4}$, then it divide the unit square into the small patches which are also square so that in this case obtain sixteen small square-patches inside the unit square with twenty-five nodes and on each node compute the corresponding value λ- value thus the λ vector display twenty-five values. On the same way when $h = \frac{1}{8}$ then divided the unit square into sixty-four small square-patches and so on. The minimum and maximum λ-values when $h = \frac{1}{4}$ is $\lambda_h = 0.847454815435$ and $\lambda_h = 1.31462192757$ respectively.
The error can be computed in L^2 norm, whose formula is defined as :

$$||\lambda - \lambda_h||_{L^2} = (\int |\lambda - \lambda_h|^2 \ dx)^{1/2}$$

$$r_\lambda = \frac{ln(E_i/E_{i-1})}{ln(h_i/h_{i-1})}$$

TABLE 3. Error Analysis

h	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.00309141295772000
1/8	0.00079836758867200
1/16	0.00020167822617300
1/32	0.00005058494827850
1/64	0.00001265907791610
1/128	0.00000297757572077

Table 4. Convergence rate

h	$r_{\lambda_{L^2}}$
1/4	1.95314
1/8	1.98500
1/16	1.99528
1/32	1.99854
1/64	2.08796

Program syntax for λ in Python
Space and boundary parts are defined same as before.
Test and trial functions are defined.

```
lam = TrialFunction(V)
v = TestFunction(V)
u-u0 = Constant("1.0")
Ve = FunctionSpace(mesh, "Lagrange" , degree=3)
lam_a = Expression("-(exp(x[1])+exp(-x[1]))/(exp(1)-exp(-1))")
lam_ext = interpolate(lam_a,Ve)
```

Variational equation is defined.

```
a = inner(grad(lam), grad(v))*dx + inner(lam,v)*dx
L = (u-u0)*v*ds(1)
```

This syntax shows how to integrate the variational equation and how to see the variational form into matrix form.

```
A = assemble(a, exterior_facet_domains=boundary_parts)
b = assemble(L, exterior_facet_domains=boundary_parts)

print'A : ', A.array()
print'b : ', b.array()
```

The solution function is defined.

```
lam = Function(V)
LAM = lam.vector()
solve(A, LAM, b)
```

The error norm is defined in L^2 space.

```
error_2 = inner(lam_ext,lam_ext)*dx
E = sqrt(assemble(error_2))
```

The convergence rate is defined.

```
from math import log
for i in range(1, len(E_1)):
r_1 = log(E_1[i]/E_1[i-1])/log(h[i]/h[i-1])
print 'cr(%.7f) : %.5f'%(h[i],r_1)
```

Choose q, $f \equiv 1$ and solve for u in strong form for exact solution. Chosen $s(u)$ as a linear functional of u such that $s(u) = u$ so that $s'(u) = 1$. Now the goal is to find exact solution when $q = 1.0$

on Γ_c and $f = 1$. Chosen such value of q find u which satisfies the boundary conditions and satisfies the equation in strong form this gives the value of f when we substitute u in the given strong form. So that we obtain following u function.

$$u = \frac{e^{(y-1)} + e^{-(y-1)}}{e - e^{-1}} + 1 = \frac{\cosh(y-1)}{\sinh(1)} + 1$$

Compute their values corresponding to various meshes. Continuously computing the values of u corresponding to different step size. The maximum and minimum u values when step size $h = \frac{1}{4}$ is $u = 2.3130352855$ and $u = 1.85091812824$ respectively. So that the maximum and minimum values of u can be obtained when $y = 0$ and $y = 1$, from the exact solution, which are the boundary parts of Γ_c and Γ_o respectively. Thus the maximum value of u lies on the boundary Γ_c where $y = 0$ and the minimum value of u lies on the boundary Γ_o where $y = 1$.
For arbitrary q the exact solution of u is :

$$u = q \cdot \frac{e^{(y-1)} + e^{-(y-1)}}{e - e^{-1}} + 1 = q \cdot \frac{\cosh(y-1)}{\sinh(1)} + 1$$

Choose q, $f \equiv 1$ and solve for u_h in variational form for approximate solution. The *finite element method* is used as an approximation method. Now, chosen $q = 1.0$ and $f = 1$ compute u_h - value by finite element method corresponding to various meshes. Thus the maximum and minimum u_h values when $h = \frac{1}{4}$ is $u_h = 2.31462192757$ and $u_h = 1.84745481543$ respectively.
To solve the variational equation used *assemble* which is integrated whole equation and it include matrix and vector.
Formulas in L^2 which are frequently used to calculate error and convergence rate corresponding to different values of step size h. To computing the error in python used L^2 norm where we used term *assemble* which is integrated over space.

$$||u - u_h||_{L^2} = (\int |u - u_h|^2 \ dx)^{1/2}$$

$$r_u = \frac{\ln(E_i/E_{i-1})}{\ln(h_i/h_{i-1})}$$

It is an error and convergence rate corresponding to various meshes and the exact solution is interpolated onto the space of degree three.The approximate solution is interpolated onto the space of degree one whereas the exact solution is interpolated onto the space of degree three. This gives better analysis.
It can be observed that error is decreasing with an approximate factor $\frac{1}{4}$ as step size h is decreasing.

TABLE 5. Error Analysis

h	$\|\|u - u_h\|\|_{L^2}$
1/4	0.00309141296018000
1/8	0.00079836759786100
1/16	0.00020167826376200
1/32	0.00005058509759330
1/64	0.00001265965512210
1/128	0.00000316878219317

TABLE 6. Convergence rate

h	$r_{u_{L^2}}$
1/4	1.95314
1/8	1.98500
1/16	1.99527
1/32	1.99847
1/64	1.99824

Choose q, f and solve for u in strong form for exact solution. Chosen $s(u)$ as a linear functional of u such that $s(u) = u$ so that $s'(u) = 1$. The aim is to find exact solution when $q = 1.0$ on Γ_c and $f = y^2 + 1$. Chosen such value of q find u which satisfies the boundary conditions and satisfies the equation in strong form this gives the value of f when we substitute u in the given strong form. So that we obtain following u function.

$$u = \frac{e^y + e^{-y}}{e + e^{-1}}(e^{-1} - 2) + e^{-y} + y^2 + 3 = \frac{\cosh(y)}{\sinh(1)}(e^{-1} - 2) + e^{-y} + y^2 + 3$$

Compute their values corresponding to various meshes. Continuously computing the values of u corresponding to different step size. The maximum and minimum u values when step size $h = \frac{1}{4}$ is $u = 2.61119902902$ and $u = 2.22484755724$ when $y = 0$ and $y = 1$ respectively. These are the boundary parts thus maximum of u occurred on the boundary Γ_c where $y = 0$ and minimum of u occurred on Γ_o where $y = 1$.

For arbitrary q the exact solution of u is :

$$u = \frac{e^y + e^{-y}}{e + e^{-1}}(e^{-1} - 2) + qe^{-y} + y^2 + 3 = \frac{\cosh(y)}{\sinh(1)}(e^{-1} - 2) + qe^{-y} + y^2 + 3$$

Choose q, f and solve for u_h in variational form for approximate solution. The *finite element method* is used as an approximation method. Now, chosen $q = 1.0$ and $f = y^2 + 1$ compute u_h - value by finite element method corresponding to various meshes. Thus the maximum and minimum u_h values when $h = \frac{1}{4}$ is $u_h = 2.62651088775$ and $u_h = 2.22558560553$ respectively.

Formulas in L^2 which are frequently used to calculate error and convergence rate corresponding to various meshes. To computing the error in python used L^2 norm where we used term *assemble* which is integrated over space.

$$||u - u_h||_{L^2} = (\int |u - u_h|^2 \, dx)^{1/2}$$

$$r_u = \frac{\ln(E_i/E_{i-1})}{\ln(h_i/h_{i-1})}$$

It is an error and convergence rate corresponding to various meshes and the exact solution is interpolated onto the space of degree three. The approximate solution is interpolated onto the space of degree one whereas the exact solution is interpolated onto the space of degree three. This gives better analysis.

It can be observed that error is decreasing with an approximate factor $\frac{1}{4}$ as step size h is decreasing.

TABLE 7. Error Analysis

h	$\|\|u - u_h\|\|_{L^2}$
1/4	0.0110396262776000
1/8	0.0027720761890800
1/16	0.0006940142979160
1/32	0.0001735807512930
1/64	0.0000434012460430
1/128	0.0000102043418957

TABLE 8. Convergence rate

h	$r_{u_{L^2}}$
1/4	1.99365
1/8	1.99793
1/16	1.99936
1/32	1.99980
1/64	2.08855

3.8. **An optimal control problem with Neumann boundary condition.** In this section solve finite element equations as one equation, this can be done by adding all finite element equations then it is an equation which contain more than one variables. In beginning solve individually. Then solve two stationary point equations as one equation. Finally solve all three finite element equations as one equation.
To solve finite element equations as one equation choose the objective function subjected to elliptic PDE's constraints with Neumann boundary conditions defined as:
minimise the objective function

$$J(u,q) = \frac{1}{2}||u-u_0||^2_{\Gamma_o} + \frac{1}{2}\alpha||q||^2_{\Omega}$$

where Γ_o, observational boundary,
subjected to constraints

$$\begin{aligned} -\nabla^2 u + u &= q \text{ in } \Omega \\ \partial_n u &= 0 \text{ on } \Gamma \end{aligned}$$

The spaces $V = H^1(\Omega)$ for state variable u and $Q = L_2(\Omega)$ for control variable q. Now the second term of cost function is defined norm of the square over the whole space therefore $q \in L_2(\Omega)$.
Then the Lagrangian is :

$$L(u,q,\lambda) = \frac{1}{2}||u-u_0||^2_{\Gamma_0} + \frac{1}{2}\alpha||q||^2_{\Omega} - (\nabla u, \nabla\lambda)_\Omega - (u,\lambda)_\Omega + (q,\lambda)_\Omega$$

Therefore equations for stationary point $x := (u,q,\lambda)$ are :

$$\begin{aligned} &L'_u(u,q,\lambda)(\varphi) = (\varphi, u-u_0)_{\Gamma_o} - (\nabla\varphi,\nabla\lambda)_\Omega - (\varphi,\lambda)_\Omega = 0 \text{ for all } \varphi \in H^1(\Omega) \\ &L'_q(u,q,\lambda)(\chi) = \alpha(\chi,q)_\Omega + (\chi,\lambda)_\Omega = 0 \text{ for all } \chi \in L_2(\Omega) \\ &L'_\lambda(u,q,\lambda)(\psi) = -(\nabla u,\nabla\psi)_\Omega - (u,\psi)_\Omega + (q,\psi)_\Omega = 0 \text{ for all } \psi \in H^1(\Omega) \end{aligned}$$

Now find the strong form to calculate the exact solution of these stationary point equations which is used to compute the error analysis.

$$(\varphi, u-u_0)_{\Gamma_o} - (\nabla\varphi,\nabla\lambda)_\Omega - (\varphi,\lambda)_\Omega = 0 \text{ for all } \varphi \in H^1(\Omega)$$

$$\boxed{\begin{array}{c} -\nabla^2\lambda + \lambda = 0 \text{ in } \Omega \\ \partial_n\lambda = u-u_0 \ \text{ on } \Gamma_o \\ \partial_n\lambda = 0 \ \text{ on } \Gamma\setminus\Gamma_o \end{array}}$$

$$\alpha(\chi,q)_\Omega + (\chi,\lambda)_\Omega = 0 \text{ for all } \chi \in L_2(\Omega)$$

$$\boxed{\alpha q + \lambda = 0 \text{ in } \Omega}$$

$$-(\nabla u,\nabla\psi)_\Omega - (u,\psi)_\Omega + (q,\psi)_\Omega = 0 \text{ for all } \psi \in H^1(\Omega)$$

$$\boxed{\begin{array}{c} -\nabla^2 u + u = q \ \text{ in } \Omega \\ \partial_n u = 0 \text{ on } \Gamma \end{array}}$$

Since α and u_0 are involved in the stationary point equations therefore the solution term contains α and u_0.

The exact solution is:

$$u = \frac{A}{\alpha}\underbrace{\left[\frac{-\cosh(y)}{2\sinh(1)}(1+\coth(1)) + \frac{y}{2}\frac{\sinh(y)}{\sinh(1)}\right]} = \frac{A}{\alpha}v(y)$$

$$q = -\frac{A}{\alpha}\frac{\cosh(y)}{\sinh(1)}$$

$$\lambda = A\frac{\cosh(y)}{\sinh(1)}$$

$$\text{where} \quad A = u\,|_{\Gamma_o} - u_0 = \frac{A}{\alpha}v(1) - u_0$$

$$\text{implies} \quad u_0 = A(\frac{v(1)-\alpha}{\alpha})$$

$$\text{so that} \quad A = \frac{\alpha u_0}{v(1)-\alpha} \quad \text{or} \quad \frac{A}{\alpha} = \frac{u_0}{v(1)-\alpha}$$

$$\text{Thus} \quad u = \frac{u_0}{v(1)-\alpha}v(y)$$

$$q = \frac{-u_0}{(v(1)-\alpha)}\frac{\cosh(y)}{\sinh(1)}$$

$$\lambda = \alpha\frac{-u_0}{(v(1)-\alpha)}\frac{\cosh(y)}{\sinh(1)}$$

The solution behaviour when $\alpha \to 0$

$$\text{Thus} \quad u = \frac{u_0}{v(1)-\alpha}v(y) \to \frac{u_0}{v(1)}v(y) \quad \text{in} \quad \Omega \text{ when } \alpha = 0$$

$$\text{and} \quad u = \frac{u_0}{v(1)-\alpha}v(y) \to u_0 \quad \text{on} \quad \Gamma_0 \text{ when } \alpha = 0$$

$$q = \frac{-u_0}{(v(1)-\alpha)}\frac{\cosh(y)}{\sinh(1)} \to \frac{-u_0}{v(1)}\frac{\cosh(y)}{\sinh(1)} \text{ when } \alpha = 0$$

$$\lambda = \alpha\frac{-u_0}{(v(1)-\alpha)}\frac{\cosh(y)}{\sinh(1)} \to 0 \text{ when } \alpha = 0$$

$$u\,|_{\Gamma_o} - u_0 = u_0\frac{v(1)}{v(1)-\alpha} - u_0 = u_0(\frac{v(1)-v(1)+\alpha}{v(1)-\alpha}) = \frac{\alpha u_0}{v(1)-\alpha}$$

$$\text{where} \quad u\,|_{\Gamma_o} = u_0\frac{v(1)}{v(1)-\alpha} = u_0\frac{v(1)-\alpha+\alpha}{v(1)-\alpha} = u_0(1+\frac{\alpha}{v(1)-\alpha}) \to u_0$$

when $\alpha = 0$

Thus $J(u,q) \to 0$ when $\alpha \to 0$ since $u \to u_0$.

This is an exact solution when solving stationary point equations individual, combine and as one equation.

3.9. **Solve finite element equations individual by FEM.** To compute the solution of these variational equations individually.
Choose u_0, q solve for u_h by finite element method. Chosen $u_0 = 1.0$, $q = \frac{-u_0}{(v(1)-\alpha)}\frac{\cosh(y)}{\sinh(1)}$ compute u_h - value by finite element method with meshes: mesh(4,4), mesh(8,8), mesh(16,16), mesh(32,32), mesh(64,64), mesh(128,128) then the error is decreasing with the factor 1/4 with this sequence. The convergence rate approaches to 2. The error formula is:

$$||u-u_h||_{L^2} = (\int |u-u_h|^2\,dx)^{1/2}$$

$$r_u = \frac{\ln(E_i/E_{i-1})}{\ln(h_i/h_{i-1})}$$

It is an error and convergence rate corresponding to various meshes. The exact solution is interpolated onto the space of degree three.The approximate solution is interpolated onto the space of degree one whereas the exact solution is interpolated onto the space of degree three. This gives better analysis.
It can be observed that error is decreasing with an approximate factor $\frac{1}{4}$ as step size h is decreasing.

Table 9. Error Analysis

h	$\|\|u - u_h\|\|_{L^2}$
1/4	0.0052025438008600
1/8	0.0013027726145900
1/16	0.0003258282857720
1/32	0.0000814656790296
1/64	0.0000203669700972

Table 10. Convergence rate

h	$r_{u_{L^2}}$
1/4	1.99763
1/8	1.99940
1/16	1.99985
1/32	1.99996
1/64	1.99978

Choose u_0, u and solve for λ_h by finite element method.
Chosen $u_0 = 1.0$, $u = \frac{u_0}{v(1)-\alpha} v(y)$ compute λ_h - value by finite element method corresponding to various meshes then the error is decreasing with the factor 1/4 with this sequence. The convergence rate approaches to 2. The error formula is:

$$||\lambda - \lambda_h||_{L^2} = (\int |\lambda - \lambda_h|^2 \; dx)^{1/2}$$

$$r_\lambda = \frac{ln(E_i/E_{i-1})}{ln(h_i/h_{i-1})}$$

It is an error and convergence rate corresponding to various meshes and the exact solution is interpolated onto the space of degree three. It can be observed that error is decreasing with a factor $\frac{1}{4}$ as step size h is decreasing.

Table 11. Error Analysis

h	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.00309141295772000
1/8	0.00079836758867200
1/16	0.00020167822617300
1/32	0.00005058494827850
1/64	0.00001265907791610
1/128	0.00000297757572077

Table 12. Convergence rate

h	$r_{\lambda_{L^2}}$
1/4	1.95314
1/8	1.98500
1/16	1.99528
1/32	1.99854
1/64	2.08796

3.10. **Solve finite element equations combine by FEM.** There is three equations for stationary point, from the q-equation substitute $q = \frac{-1}{\alpha}\lambda$ into λ-equation then the stationary point equations become:

$$
\begin{aligned}
(\varphi, u)_{\Gamma_o} \quad &- (\nabla\varphi, \nabla\lambda)_\Omega - (\varphi, \lambda)_\Omega = (\varphi, u_0)_{\Gamma_o} \quad \text{for all } \varphi \in H^1(\Omega) \\
-(\nabla u, \nabla\psi)_\Omega - (u, \psi)_\Omega \quad &- \quad \frac{1}{\alpha}(\lambda, \psi)_\Omega = \quad 0 \qquad \text{for all } \psi \in H^1(\Omega)
\end{aligned}
$$

solve these equations for u, λ keeping u_0 known for various α's and compute error, convergence rate. equations in matrix form

$$
\begin{aligned}
A_{\Gamma_o}^T\bar{u} \; - \quad &(B_\Omega^T + A_\Omega^T)\bar{\lambda} = A_{\Gamma_o}^T u_0 \\
-(B_\Omega + A_\Omega)\bar{u} \quad - \quad &\frac{1}{\alpha}A_\Omega\bar{\lambda} \quad = 0
\end{aligned}
$$

into matrix form as follows:

$$
\begin{bmatrix} A_{\Gamma_o}^T & -(B_\Omega^T + A_\Omega^T) \\ -(B_\Omega + A_\Omega) & \frac{-1}{\alpha}A_\Omega \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} A_{\Gamma_o}^T u_0 \\ 0 \end{bmatrix}
$$

where the square matrix has eigenvalues $\frac{(A_{\Gamma_o} - \frac{1}{\alpha}A_\Omega)}{2} \pm \sqrt{\frac{(A_{\Gamma_o} - \frac{1}{\alpha}A_\Omega)^2}{4} + ((B_\Omega + A_\Omega)(B_\Omega^T + A_\Omega^T) + \frac{1}{\alpha}A_\Omega A_{\Gamma_o})}$, one is positive, one is negative, therefore it is a saddle point.

The numerical form of the matrices when $\alpha = 1$ with mesh $(2,1)$:

$$A_{\Gamma_o}^T = \begin{bmatrix} .5 & .083 & 0 & .167 & 0 & 0 \\ .083 & .33 & .083 & 0 & 0 & 0 \\ 0 & .083 & .5 & 0 & 0 & .167 \\ .167 & 0 & 0 & .33 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .167 & 0 & 0 & .33 \end{bmatrix}, \quad A_{\Gamma_o}^T u_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ .25 \\ .5 \\ .25 \end{bmatrix}$$

$$-(B_\Omega + A_\Omega) = \begin{bmatrix} -1.3 & .98 & 0 & .23 & -.042 & 0 \\ .98 & -2.6 & .98 & 0 & .458 & -.042 \\ 0 & .98 & -1.3 & 0 & 0 & .23 \\ .23 & 0 & 0 & -1.3 & .98 & 0 \\ -.042 & .458 & 0 & .98 & -2.6 & .98 \\ 0 & -.042 & .23 & 0 & .98 & -1.3 \end{bmatrix},$$

$$-(B_\Omega^T + A_\Omega^T) = \begin{bmatrix} -1.3 & .98 & 0 & .23 & -.042 & 0 \\ .98 & -2.6 & .98 & 0 & .458 & -.042 \\ 0 & .98 & -1.3 & 0 & 0 & .23 \\ .23 & 0 & 0 & -1.3 & .98 & 0 \\ -.042 & .458 & 0 & .98 & -2.6 & .98 \\ 0 & -.042 & .23 & 0 & .98 & -1.3 \end{bmatrix},$$

$-(B_\Omega^T + A_\Omega^T)$ is transpose of $-(B_\Omega + A_\Omega)$.

$$\frac{-1}{\alpha} A_\Omega = \begin{bmatrix} .083 & -.0208 & 0 & -.0208 & -.042 & 0 \\ -.0208 & -.125 & -.0208 & 0 & -.042 & -.042 \\ 0 & -.0208 & -.042 & 0 & 0 & -.0208 \\ -.0208 & 0 & 0 & -.042 & -.0208 & 0 \\ -.042 & -.042 & 0 & -.0208 & -.125 & -.0208 \\ 0 & -.042 & -.0208 & 0 & -.0208 & -.083 \end{bmatrix}$$

the matrix form

$$\begin{bmatrix} A_{\Gamma_o}^T & -(B_\Omega^T + A_\Omega^T) \\ -(B_\Omega + A_\Omega) & \frac{-1}{\alpha} A_\Omega \end{bmatrix} =$$

$$\begin{bmatrix} .5 & .083 & 0 & .167 & 0 & 0 & -1.3 & .98 & 0 & .23 & -.042 & 0 \\ .083 & .33 & .083 & 0 & 0 & 0 & .98 & -2.6 & .98 & 0 & .458 & -.042 \\ 0 & .083 & .5 & 0 & 0 & .167 & 0 & .98 & -1.3 & 0 & 0 & .23 \\ .167 & 0 & 0 & .33 & 0 & 0 & .23 & 0 & 0 & -1.3 & .98 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.042 & .458 & 0 & .98 & -2.6 & .98 \\ 0 & 0 & .167 & 0 & 0 & .33 & 0 & -.042 & .23 & 0 & .98 & -1.3 \\ -1.3 & .98 & 0 & .23 & -.042 & 0 & .083 & -.0208 & 0 & -.0208 & -.042 & 0 \\ .98 & -2.6 & .98 & 0 & .458 & -.042 & -.0208 & -.125 & -.0208 & 0 & -.042 & -.042 \\ 0 & .98 & -1.3 & 0 & 0 & .23 & 0 & -.0208 & -.042 & 0 & 0 & -.0208 \\ .23 & 0 & 0 & -1.3 & .98 & 0 & -.0208 & 0 & 0 & -.042 & -.0208 & 0 \\ -.042 & .458 & 0 & .98 & -2.6 & .98 & -.042 & -.042 & 0 & -.0208 & -.125 & -.0208 \\ 0 & -.042 & .23 & 0 & .98 & -1.3 & 0 & -.042 & -.0208 & 0 & -.0208 & -.083 \end{bmatrix}$$

The error analysis corresponding to $\alpha = 1.0$ shows that error is decreasing with factor $\frac{1}{4}$ as meshes are decreasing and convergence rate is approach to 2.

TABLE 13. Error analysis, $\alpha = 1.0$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.0001271871614630000	0.002802482408900000
1/8	0.0000339252793065000	0.000706006955695000
1/16	0.0000086461244033000	0.000176929218032000
1/32	0.0000021742079668800	0.000044265500688500
1/64	0.0000005442057870710	0.000011068859373600
1/128	0.0000001348049496360	0.000002767334468500
1/256	0.0000000278781655806	0.000000691581891606

TABLE 14. Convergence rate

h	r_u	r_λ
1/4	1.90652	1.98895
1/8	1.97224	1.99651
1/16	1.99156	1.99892
1/32	1.99827	1.99968
1/64	2.01328	1.99994
1/128	2.27367	2.00053

The error analysis corresponding to $\alpha = \frac{1}{10}$ shows that error is decreasing with factor $\frac{1}{4}$ as meshes are decreasing.

TABLE 15. Error analysis, $\alpha = \frac{1}{10}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.0002531035992230000	0.000515429710174000
1/8	0.0000679699767978000	0.000130409786850000
1/16	0.0000173809300207000	0.000032733429784100
1/32	0.0000043761860599600	0.000008193599351780
1/64	0.0000010959389391700	0.000002049155396410
1/128	0.0000002719908078760	0.000000512332910811
1/256	0.0000000587876159303	0.000000128045530736

TABLE 16. Convergence rate

h	r_u	r_λ
1/4	1.89676	1.98272
1/8	1.96739	1.99422
1/16	1.98976	1.99820
1/32	1.99751	1.99947
1/64	2.01054	1.99988
1/128	2.20997	2.00042

The error analysis corresponding to $\alpha = \frac{1}{100}$ shows that error is decreasing with factor $\frac{1}{4}$ as meshes are decreasing and convergence rate is approach to 2.

TABLE 17. Error analysis, $\alpha = \frac{1}{100}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.0002974550387080000	0.0000594440750513000
1/8	0.0000821499048279000	0.0000153683747916000
1/16	0.0000212032218845000	0.0000038950058119800
1/32	0.0000053529132585400	0.0000009782954445650
1/64	0.0000013416303070500	0.0000002449098835350
1/128	0.0000003336860802280	0.0000000612494781019
1/256	0.0000000748066392797	0.0000000153098376669

TABLE 18. Convergence rate

h	r_u	r_λ
1/4	1.85634	1.95157
1/8	1.95398	1.98027
1/16	1.98589	1.99328
1/32	1.99634	1.99802
1/64	2.00742	1.99948
1/128	2.15725	2.00024

The error analysis corresponding to $\alpha = \frac{1}{1000}$ shows that error is decreasing with factor $\frac{1}{4}$ as meshes are decreasing.

TABLE 19. Error analysis, $\alpha = \frac{1}{1000}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.000637322527393000	0.00000463085481704000
1/8	0.000171552692575000	0.00000149737518910000
1/16	0.000043798009532400	0.00000043162101589700
1/32	0.000011016535072700	0.00000011402025288900
1/64	0.000002759927901520	0.00000002900370855020
1/128	0.000000693597883475	0.00000000728634341007
1/256	0.000000185977869469	0.00000000182513611470

TABLE 20. Convergence rate

h	r_u	r_λ
1/4	1.82426	1.79964
1/8	1.93295	1.86887
1/16	1.97641	1.94561
1/32	1.99299	1.98250
1/64	2.00511	1.99511
1/128	2.12707	1.99891

3.11. **Solve finite element equations as one equation by FEM.** Matrix form of stationary point equations

$$
\begin{array}{ccccccccl}
(\varphi, u)_{\Gamma_o} & + & 0 & - (\nabla\varphi, \nabla\lambda)_\Omega - (\varphi, \lambda)_\Omega & = & (\varphi, u_0)_{\Gamma_o} & \text{for all } \varphi \in H^1(\Omega) \\
0 & + & \alpha(\chi, q)_\Omega & + \quad (\chi, \lambda)_\Omega & = & 0 & \text{for all } \chi \in L_2(\Omega) \\
-(\nabla u, \nabla\psi)_\Omega - (u, \psi)_\Omega & + & (q, \psi)_\Omega & + \quad 0 & = & 0 & \text{for all } \psi \in H^1(\Omega)
\end{array}
$$

equations in matrix form

$$
\begin{array}{rcccl}
A_{\Gamma_o}^T \bar{u} + & 0 & - & (B_\Omega^T + A_\Omega^T)\bar{\lambda} & = A_{\Gamma_o}^T u_0 \\
0 \quad + & \alpha A_\Omega^T \bar{q} & + & A_\Omega^T \bar{\lambda} & = 0 \\
-(B_\Omega + A_\Omega)\bar{u} + & A_\Omega \bar{q} & + & 0 & = 0
\end{array}
$$

into matrix form as follows:

$$
\left[\begin{array}{c|c|c}
A_{\Gamma_o}^T & 0 & -(B_\Omega^T + A_\Omega^T) \\ \hline
0 & \alpha A_\Omega^T & A_\Omega^T \\ \hline
-(B_\Omega + A_\Omega) & A_\Omega & 0
\end{array}\right]
\begin{bmatrix} u \\ q \\ \lambda \end{bmatrix}
=
\begin{bmatrix} A_{\Gamma_o}^T u_0 \\ 0 \\ 0 \end{bmatrix}
$$

the block form of the matrix is :

$$
\begin{bmatrix} A_e & B_e^T \\ B_e & 0 \end{bmatrix}
$$

where

$$
A_e = \begin{bmatrix} A_{\Gamma_o}^T & 0 \\ 0 & \alpha A_\Omega^T \end{bmatrix}, B_e = \begin{bmatrix} -(B_\Omega + A_\Omega) & A_\Omega \end{bmatrix},
$$

$$
B_e^T = \begin{bmatrix} -(B_\Omega^T + A_\Omega^T) \\ A_\Omega^T \end{bmatrix}, b_e = \begin{bmatrix} A_{\Gamma_o}^T u_0^T \end{bmatrix}
$$

Thus the block form of the matrix has eigenvalues $\frac{A_e}{2} \pm \sqrt{\frac{{A_e}^2}{4} + B_e B_e^T}$, one positive, one negative, therefore it is a saddle point.

Numerical form of the matrices, when $\alpha = 1$ with mesh $(2,1)$

$$A_e = \begin{bmatrix} A_{\Gamma_o}^T & 0 \\ 0 & \alpha A_\Omega^T \end{bmatrix} = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c}
.5 & .083 & 0 & .167 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
.083 & .33 & .083 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & .083 & .5 & 0 & 0 & .167 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
.167 & 0 & 0 & .33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & .167 & 0 & 0 & .33 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & .083 & .0208 & 0 & .0208 & .042 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & .0208 & .125 & .0208 & 0 & .042 & .042 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & .0208 & .042 & 0 & 0 & .0208 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & .0208 & 0 & 0 & .042 & .0208 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & .042 & .042 & 0 & .0208 & .125 & .0208 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & .042 & .0208 & 0 & .0208 & .083
\end{array}\right]$$

A_e is positive definite.

$$B_e = \begin{bmatrix} -(B_\Omega + A_\Omega) & A_\Omega \end{bmatrix} = \left[\begin{array}{cccccc|cccccc}
-1.3 & .98 & 0 & .23 & -.042 & 0 & .083 & .0208 & 0 & .0208 & .042 & 0 \\ \hline
.98 & -2.6 & .98 & 0 & .45 & -.042 & .0208 & .125 & .0208 & 0 & .042 & .042 \\ \hline
0 & .98 & -1.3 & 0 & 0 & .23 & 0 & .0208 & .042 & 0 & 0 & .0208 \\ \hline
.23 & 0 & 0 & -1.3 & .98 & 0 & .0208 & 0 & 0 & .042 & .0208 & 0 \\ \hline
-.042 & .4583 & 0 & .98 & -2.6 & .98 & .042 & .042 & 0 & .0208 & .125 & .0208 \\ \hline
0 & -.042 & .23 & 0 & .98 & -1.3 & 0 & .42 & .0208 & 0 & .0208 & .083
\end{array}\right]$$

$$B_e^T = \begin{bmatrix} -(B_\Omega^T + A_\Omega^T) \\ A_\Omega^T \end{bmatrix} = \left[\begin{array}{c|c|c|c|c|c}
-1.33 & .98 & 0 & .23 & -.042 & 0 \\
.98 & -2.625 & .98 & 0 & .4583 & -.042 \\
0 & .98 & -1.3 & 0 & 0 & .23 \\
.23 & 0 & 0 & -1.3 & .98 & 0 \\
-.042 & .4583 & 0 & .98 & -2.625 & .98 \\
0 & -.04167 & .23 & 0 & .98 & -1.33 \\ \hline
.0833 & .02083 & 0 & .02083 & .04167 & 0 \\
.02083 & .125 & .02083 & 0 & .042 & .042 \\
0 & .02083 & .04167 & 0 & 0 & .02083 \\
.02083 & 0 & 0 & .04167 & .02083 & 0 \\
.042 & .042 & 0 & .02083 & .125 & 0.02083 \\
0 & .04167 & .02083 & 0 & .02083 & .0833
\end{array}\right], b_e = \begin{bmatrix} A_{\Gamma_o}^T u_0^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.25 \\ 0.5 \\ 0.25 \end{bmatrix}$$

B_e^T is the transport of B_e.

Add these three equations into one equation :

$$(\nabla\psi, \nabla u)_\Omega + (\nabla\varphi, \nabla\lambda)_\Omega + (\psi, u)_\Omega + (\varphi, \lambda)_\Omega + \alpha(\chi, q)_\Omega + (\chi, \lambda)_\Omega - (\psi, q)_\Omega - (\varphi, u)_{\Gamma_o}$$
$$= -(\varphi, u_0)_{\Gamma_o} \text{ for all } \psi, \varphi \in V \text{ and } \chi \in Q$$

Finite element equation :

$$(\nabla\psi_h, \nabla u_h)_\Omega + (\nabla\varphi_h, \nabla\lambda_h)_\Omega + (\psi_h, u_h)_\Omega + (\varphi_h, \lambda_h)_\Omega + \alpha(\chi_h, q_h)_\Omega + (\chi_h, \lambda_h)_\Omega - (\psi_h, q_h)_\Omega - (\varphi_h, u_h)_{\Gamma_o}$$
$$= -(\varphi_h, u_0)_{\Gamma_o} \text{ for all } \psi_h, \varphi_h \in V_h \text{ and } \chi_h \in Q_h$$

Choose $u_0 = 1$, keeping (u_h, q_h, λ) are variables.

Compute the objective function corresponding to various meshes and α values, it shows that value of objective function is decreasing as α is decreasing.

TABLE 21. Objective function for various α, $h = \frac{1}{4}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50874855461900	0.1249892620210000	0.63373781664000
1/10	0.02261271860870	0.0406994483497000	0.06331216695850
1/100	0.00189580623560	0.0048132555633400	0.00670906179894
1/1000	0.00111277114959	0.0004898622862820	0.00160263343588
1/10000	0.00104509048393	0.0000490747675328	0.00109416525146

TABLE 22. Objective function for various α, $h = \frac{1}{8}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50869662890400	0.1249893964970000	0.63368602540100
1/10	0.02258018653160	0.0407030890363000	0.06328327556790
1/100	0.00188998612035	0.0048137626616000	0.00670374878195
1/1000	0.00111186870246	0.0004899125415890	0.00160178124405
1/10000	0.00104835647857	0.0000490782205093	0.00109743469907

TABLE 23. Objective function for various α, $h = \frac{1}{16}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50868222633200	0.124989432880000	0.63367165921300
1/10	0.02257100503590	0.040704112038600	0.06327511707450
1/100	0.00188817453997	0.004813906416100	0.00670208095607
1/1000	0.00111135151714	0.000489927001483	0.00160127851863
1/10000	0.00104913644090	0.000049079349339	0.00109821579024

TABLE 24. Objective function for various α, $h = \frac{1}{32}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50867854031900	0.1249894421440000	0.63366798246300
1/10	0.02256862517100	0.0407043783410000	0.06327300351190
1/100	0.00188768288631	0.0048139439694200	0.00670162685573
1/1000	0.00111114834062	0.0004899308469340	0.00160107918756
1/10000	0.00104912452421	0.0000490796990666	0.00109820422328

The objective function value is decreasing as α is decreasing corresponding to various meshes.

Table 25. Objective function for various α, $h = \frac{1}{64}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50867854031900	0.1249894421440000	0.63366798246300
1/10	0.02256862517100	0.0407043783410000	0.06327300351190
1/100	0.00188768288631	0.0048139439694200	0.00670162685573
1/1000	0.00111114834062	0.0004899308469340	0.00160107918756
1/10000	0.00104912452421	0.0000490796990666	0.00109820422328

Table 26. Objective function for various α,$h = \frac{1}{128}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50867854031900	0.1249894421440000	0.63366798246300
1/10	0.02256862517100	0.0407043783410000	0.06327300351190
1/100	0.00188768288631	0.0048139439694200	0.00670162685573
1/1000	0.00111114834062	0.0004899308469340	0.00160107918756
1/10000	0.00104912452421	0.0000490796990666	0.00109820422328

The analysis of an objective function value corresponding to exact solution and finite element solution when mesh is unitsquare, $h = \frac{1}{256}$.

Table 27. Objective function for various α, $h = \frac{1}{256}$

α	$\frac{1}{2}\|\|u_h - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q_h\|\|^2$	$J(u_h, q_h)$
1.0	0.50867854031900	0.1249894421440000	0.63366798246300
1/10	0.02256862517100	0.0407043783410000	0.06327300351190
1/100	0.00188768288631	0.0048139439694200	0.00670162685573
1/1000	0.00111114834062	0.0004899308469340	0.00160107918756
1/10000	0.00104912452421	0.0000490796990666	0.00109820422328

Table 28. Objective function for various α, exact solution

α	$\frac{1}{2}\|\|u - u_0\|\|^2$	$\frac{\alpha}{2}\|\|q\|\|^2$	$J(u, q)$
1.0	0.50867115387900	0.1262912256430000	0.63496237952300
1/10	0.02255557152540	0.0411284104767000	0.06368398200210
1/100	0.00187394535328	0.0048640946393100	0.00673803999259
1/1000	0.00109735179509	0.0004950348646080	0.00159238665970
1/10000	0.00103533968428	0.0000495910001043	0.00108493068438

The error analysis which is decreasing with the factor $\frac{1}{4}$ as meshes are decreasing when $\alpha = 1.0$ and convergence rate approaches to 2. The maximum and minimum vales of exact solution and finite element solution are also computed.

Table 29. Error analysis, $\alpha = 1.0$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|q - q_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.0002990139978790000	0.001536071228080000	0.001536071228090000
1/8	0.0000798812633927000	0.000397157961062000	0.000397157961068000
1/16	0.0000204015643923000	0.000100367790088000	0.000100367790120000
1/32	0.0000051352422770300	0.000025177218000900	0.000025177217635700
1/64	0.0000012869490828300	0.000006300890230960	0.000006300890403840
1/128	0.0000003237313962400	0.000001576074300580	0.000001576074506140
1/256	0.0000000876797923794	0.000000395537260241	0.000000395533660406

Table 30. Convergence rate, $\alpha = 1.0$

h	r_u	r_q	r_λ
1/4	1.90428	1.99445	1.99446
1/8	1.96918	1.99445	1.99446
1/16	1.99018	1.99445	1.99446
1/32	1.99648	1.99445	1.99446
1/64	1.99109	1.99445	1.99446
1/128	1.88448	1.99445	1.99446

Table 31. maximum minimum, $\alpha = 1.0$

u_{max}	0.504594507756	u_{min}	0.487529450441	$u_{h_{max}}$	0.504595074803	$u_{h_{min}}$	0.487529082467
q_{max}	0.650484891946	q_{min}	0.421549514180	$q_{h_{max}}$	0.650487490162	$q_{h_{min}}$	0.421546641055
λ_{max}	$-.421546641055$	λ_{min}	$-.421546641055$	$\lambda_{h_{max}}$	$-.650484891946$	$\lambda_{h_{min}}$	$-.650487490162$

The error analysis which is decreasing with the factor $\frac{1}{4}$ as meshes are decreasing when $\alpha = \frac{1}{10}$ and convergence rate approaches to 2. The maximum and minimum vales of exact solution and finite element solution are also computed.

TABLE 32. Error analysis, $\alpha = \frac{1}{10}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|q - q_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.000550691579627000	0.002852948610840000	0.0002852948610910000
1/8	0.000147342490461000	0.000745085805927000	0.0000745085805906000
1/16	0.000037650975671200	0.000188959906327000	0.0000188959905961000
1/32	0.000009478721465580	0.000047450384666400	0.0000047450385350900
1/64	0.000002375556289880	0.000011878365308300	0.0000011878364512800
1/128	0.000000597396669140	0.000002971344044520	0.0000002971356007430
1/256	0.000000161437656833	0.000000745444750956	0.0000000745461756646

TABLE 33. Convergence rate, $\alpha = \frac{1}{10}$

h	r_u	r_q	r_λ
1/4	1.90207	1.99494	1.99492
1/8	1.96841	1.99494	1.99492
1/16	1.98992	1.99494	1.99492
1/32	1.99643	1.99494	1.99492
1/64	1.99150	1.99494	1.99492
1/128	1.88771	1.99494	1.99492

TABLE 34. maximum minimum, $\alpha = \frac{1}{10}$

u_{max}	0.910598420727	u_{min}	0.879802575744	$u_{h_{max}}$	0.910599416268	$u_{h_{min}}$	0.879801900082
q_{max}	1.173874281640	q_{min}	0.760734244963	$q_{h_{max}}$	1.173879408070	$q_{h_{min}}$	0.760729065000
λ_{max}	$-.0760734244963$	λ_{min}	$-.1173874281640$	$\lambda_{h_{max}}$	$-.076072906500$	$\lambda_{h_{min}}$	$-.1173879408070$

The error analysis which is decreasing with the factor $\frac{1}{4}$ as meshes are decreasing when $\alpha = \frac{1}{100}$ and convergence rate approaches to 2. The maximum and minimum vales of exact solution and finite element solution are also computed.

TABLE 35. Error analysis, $\alpha = \frac{1}{100}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|q - q_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.000612691721161000	0.00364821498667000	0.0000364821498664000
1/8	0.000164627993022000	0.00099791269859500	0.0000099791269869400
1/16	0.000042102761465300	0.00025811256038500	0.0000025811256080000
1/32	0.000010600388451500	0.00006525930740900	0.0000006525930796660
1/64	0.000002656633121960	0.00001636891279790	0.0000001636891436930
1/128	0.000000667867332741	0.00000409646402901	0.0000000409647290214
1/256	0.000000179782478712	0.00000102653906671	0.0000000102657778599

TABLE 36. Convergence rate, $\alpha = \frac{1}{100}$

h	r_u	r_q	r_λ
1/4	1.89595	1.99659	1.99654
1/8	1.96722	1.99659	1.99654
1/16	1.98980	1.99659	1.99654
1/32	1.99645	1.99659	1.99654
1/64	1.99197	1.99659	1.99654
1/128	1.89331	1.99659	1.99654

TABLE 37. maximum minimum, $\alpha = \frac{1}{100}$

u_{max}	0.990277560795	u_{min}	0.956787019236	$u_{h_{max}}$	0.990278581689	$u_{h_{min}}$	0.956786297473
q_{max}	1.276590573680	q_{min}	0.827299976991	$q_{h_{max}}$	1.276599520170	$q_{h_{min}}$	0.827294423161
λ_{max}	$-.0082729997699$	λ_{min}	$-.0127659057368$	$\lambda_{h_{max}}$	$-.0082729442316$	$\lambda_{h_{min}}$	$-.0127659952017$

The error analysis which is decreasing with the factor $\frac{1}{4}$ as meshes are decreasing when $\alpha = \frac{1}{1000}$ and convergence rate approaches is computed corresponding to various meshes. The maximum and minimum vales of exact solution and finite element solution are also computed.

TABLE 38. Error analysis, $\alpha = \frac{1}{1000}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|q - q_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.000637322527393000	0.00463085481705000	0.00000463085481704000
1/8	0.000171552692575000	0.00149737518918000	0.00000149737518910000
1/16	0.000043798009532400	0.00043162101610400	0.00000043162101589700
1/32	0.000011016535072700	0.00011402025218800	0.00000011402025288900
1/64	0.000002759927901520	0.00002900370742060	0.00000002900370855020
1/128	0.000000693597883475	0.00000728634299768	0.00000000728634341007
1/256	0.000000185977869469	0.00000182515235056	0.00000000182513611470

TABLE 39. Convergence rate, $\alpha = \frac{1}{1000}$

h	r_u	r_q	r_λ
1/4	1.89337	1.99718	1.99719
1/8	1.96971	1.99718	1.99719
1/16	1.99119	1.99718	1.99719
1/32	1.99697	1.99718	1.99719
1/64	1.99246	1.99718	1.99719
1/128	1.89897	1.99718	1.99719

TABLE 40. maximum minimum, $\alpha = \frac{1}{1000}$

u_{max}	0.999019173658	u_{min}	0.965232996450	$u_{h_{max}}$	0.999020119312	$u_{h_{min}}$	0.965232280315
q_{max}	1.287859596650	q_{min}	0.834602915486	$q_{h_{max}}$	1.287888280810	$q_{h_{min}}$	0.834597375314
λ_{max}	$-.0008346029155$	λ_{min}	$-.0012878595970$	$\lambda_{h_{max}}$	$-.0008345973753$	$\lambda_{h_{min}}$	$-.0012878882800$

The error analysis which is decreasing with the factor $\frac{1}{4}$ as meshes are decreasing when $\alpha = \frac{1}{10000}$ and convergence rate approaches to 2. The maximum and minimum vales of exact solution and finite element solution are also computed.

TABLE 41. Error analysis, $\alpha = \frac{1}{10000}$

h	$\|\|u - u_h\|\|_{L^2}$	$\|\|q - q_h\|\|_{L^2}$	$\|\|\lambda - \lambda_h\|\|_{L^2}$
1/4	0.000656583809315000	0.00792439248533000	0.000000792439248530000
1/8	0.000176517793290000	0.00313694706603000	0.000000313694706598000
1/16	0.000044576129667400	0.00100491982123000	0.000000100491982128000
1/32	0.000011157385220000	0.00028447625215900	0.000000028447625215500
1/64	0.000002790863021200	0.00007461046472560	0.000000007461046425120
1/128	0.000000700967140748	0.00001893539957210	0.000000001893539948760
1/256	0.000000187712216008	0.00000475409119294	0.000000000475410153835

TABLE 42. Convergence rate, $\alpha = \frac{1}{10000}$

h	r_u	r_q	r_λ
1/4	1.89517	1.99384	1.99384
1/8	1.98547	1.99384	1.99384
1/16	1.99827	1.99384	1.99384
1/32	1.99922	1.99384	1.99384
1/64	1.99329	1.99384	1.99384
1/128	1.90082	1.99384	1.99384

TABLE 43. maximum minimum, $\alpha = \frac{1}{10000}$

u_{max}	0.999901830707	u_{min}	0.966085802614	$u_{h_{max}}$	0.999902654089	$u_{h_{min}}$	0.966085088075
q_{max}	1.288997451040	q_{min}	0.835340306886	$q_{h_{max}}$	1.289132421250	$q_{h_{min}}$	0.835334773992
λ_{max}	$-.00008353403069$	λ_{min}	$-.0001288997451$	$\lambda_{h_{max}}$	$-.0000835334774$	$\lambda_{h_{min}}$	$-.0001289132421$

The error analysis of objective function which is decreasing with an approximate factor $\frac{1}{4}$ as meshes are decreasing corresponding to various α values when $\alpha = 1.0, \frac{1}{10}$. The convergence rate approaches to 2.

Table 44. Error analysis of objective function, $\alpha = 1.0$

h	$\lvert J(u,q) - J(u_h, q_h)\rvert$
1/4	0.0000706156819594000
1/8	0.0000192465649741000
1/16	0.0000049067991299000
1/32	0.0000012317015266600
1/64	0.0000003081580458590
1/128	0.0000000770289696472
1/256	0.0000000191753701806

Table 45. Convergence rate, $\alpha = 1.0$

h	r_J
1/4	1.87539
1/8	1.97175
1/16	1.99413
1/32	1.99891
1/64	2.00020
1/128	2.00615

Table 46. Error analysis of objective function, $\alpha = \frac{1}{10}$

h	$\lvert J(u,q) - J(u_h, q_h)\rvert$
1/4	0.0000397156637255000
1/8	0.0000109744633507000
1/16	0.0000028253648572500
1/32	0.0000007123896254710
1/64	0.0000001785296233760
1/128	0.0000000446597971382
1/256	0.0000000111552105331

Table 47. Convergence rate, $\alpha = \frac{1}{10}$

h	r_J
1/4	1.85556
1/8	1.95764
1/16	1.98770
1/32	1.99650
1/64	1.99911
1/128	2.00126

The error analysis of objective function corresponding to exact solution and finite element solution, which is decreasing with the factor $\frac{1}{4}$ as meshes are decreasing, corresponding to various α values when $\alpha = \frac{1}{100}, \frac{1}{1000}$. The convergence rate is also computed.

TABLE 48. Error analysis of objective function, $\alpha = \frac{1}{100}$

h	$\lvert J(u,q) - J(u_h, q_h)\rvert$
1/4	0.0000075562266102300
1/8	0.0000022760160766900
1/16	0.0000006102356031690
1/32	0.0000001562630107700
1/64	0.0000000393580998539
1/128	0.0000000098605837314
1/256	0.0000000024653309801

TABLE 49. Convergence rate, $\alpha = \frac{1}{100}$

h	r_J
1/4	1.73116
1/8	1.89907
1/16	1.96539
1/32	1.98924
1/64	1.99692
1/128	1.99989

TABLE 50. Error analysis of objective function, $\alpha = \frac{1}{1000}$

h	$\lvert J(u,q) - J(u_h, q_h)\rvert$
1/4	0.0000016141133452100
1/8	0.0000007805711650300
1/16	0.0000002790047891500
1/32	0.0000000797460463851
1/64	0.0000000208206915799
1/128	0.0000000052718530190
1/256	0.0000000013222023038

TABLE 51. Convergence rate, $\alpha = \frac{1}{1000}$

h	r_J
1/4	1.04814
1/8	1.48424
1/16	1.80680
1/32	1.93740
1/64	1.98164
1/128	1.99537

3.12. **Python program.** Dolfin contains Python library, used to write finite element programs.

```
from dolfin import*
```

define the mesh as unitsquare.

```
#choose mesh: (4,4),(8,8),(16,16),(32,32),(64,64),(128,128),(256,256)
mesh=UnitSquare(256,256)
```

Specify the spaces $X_h := Q_h \times V_h \times Q_h$.

```
U=FunctionSpace(mesh, "Lagrange", 1)
V=FunctionSpace(mesh, "Lagrange", 1)
W=FunctionSpace(mesh, "Lagrange", 1)
M=MixedFunctionSpace([U,V,W])
```

define the boundary parts, observational boundary Γ_o and $\Gamma \backslash \Gamma_o$.

```
boundary_parts = MeshFunction("uint", mesh,1)
tol = 1E-14
class UpperBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and abs(x[1]-1) < tol
Gamma_c = UpperBoundary()
Gamma_c.mark(boundary_parts, 1)
class RestBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and abs(x[0]) < tol and abs(x[1]-1) < tol and abs(x[0]-1) < tol
Gamma_t_c = RestBoundary()
Gamma_t_c.mark(boundary_parts, 2)
```

To define test, trial and constant functions.

```
(u,q, lam)=TrialFunctions(M)
(v_u,v_q, v_lam)=TestFunctions(M)
u0=Constant("1.0")
alpha=(1.0/1000)
```

defined finite element equation.

```
a=-inner(grad(lam),grad(v_u))*dx - inner(grad(u),grad(v_lam))*dx-\
   inner(lam,v_u)*dx -inner(u,v_lam)*dx+q*v_lam*dx+\
alpha*(q*v_q)*dx+(lam*v_q)*dx +u*v_u*ds(1)
L= u0*v_u*ds(1)
```

To define matrix form of finite element equations.

```
#choose mesh=UnitSquare(2,1) matrices
A=assemble(a,exterior_facet_domains=boundary_parts)
Aa=assemble(alpha*(q*v_q)*dx+u*v_u*ds,exterior_facet_domains=boundary_parts)
B=assemble(-inner(grad(u),grad(v_lam))*dx-inner(u,v_lam)*dx+q*v_lam*dx,exterior_facet_domains=\
boundary_parts)
BT=assemble(-inner(grad(lam),grad(v_u))*dx-inner(lam,v_u)*dx+(lam*v_q)*dx,exterior_facet_domains=\
boundary_parts)
b = assemble(L,exterior_facet_domains=boundary_parts)
```

To print the finite element equations into matrix form.

```
print'A :', A.array()
print 'Aa:',Aa.array()
print 'B:',B.array()
print 'Bt:',BT.array()
print'L:',b.array()
```

Define the finite element solution, split solution variables into (u_h, q_h, λ_h).

```
solution=Function(M)
s=solution.vector()
solve(A,s,b)
u,q,lam=solution.split()

#finite element solution
uh=interpolate(u,U)
qh=interpolate(q,V)
lamh=interpolate(lam,W)
```

Define the exact solution u.

```
from math import cosh,sinh
Ue=FunctionSpace(mesh, "Lagrange", 3)
u_e=Expression("(u0/(v1-alph))*(((1+(cosh(1)/sinh(1)))*(-cosh(x[1])/(2*sinh(1))))+\
(x[1]*sinh(x[1])/(2*sinh(1))))",u0=1.0,v1=(((((1+((exp(1)+exp(-1))/(exp(1)-exp(-1))))*\
((-exp(1)-exp(-1))/(2*exp(1)-2*exp(-1)))))+(1*(exp(1)-exp(-1))/(2*exp(1)-2*exp(-1))))),alph=(1.0/1000))
u_ext=interpolate(u_e,Ue)
print'u:',u_ext.vector().array()
```

Define the exact solution q.

```
Ve=FunctionSpace(mesh, "Lagrange", 3)
q_e=Expression("(-u0/(v1-alph))*(cosh(x[1])/sinh(1))",u0=1.0,v1=(((1+(cosh(1)/sinh(1)))*\
(-cosh(1)/(2*sinh(1))))+(1*sinh(1)/(2*sinh(1)))),alph=(1.0/1000))
q_ext=interpolate(q_e,Ve)
print'q:',q_ext.vector().array()
```

Define the exact solution λ.

```
We=FunctionSpace(mesh, "Lagrange", 3)
lam_e=Expression("(alph*u0/(v1-alph))*cosh(x[1])/sinh(1)",u0=1.0,v1=(((1+(cosh(1)/sinh(1)))*\
(-cosh(1)/(2*sinh(1))))+(1*sinh(1)/(2*sinh(1)))),alph=(1.0/1000))
lam_ext=interpolate(lam_e,We)
print'lam:',lam_ext.vector().array()
```

Print the finite element solution (u_h, q_h, λ_h).

```
#print (u.h,q.h,lam.h)
print'u.h:',uh.vector().array()
print'q.h:',qh.vector().array()
print'lam.h:',lamh.vector().array()
```

Compute the error norm into the space L^2.

```
#error formula ||u-u.h||
error_u = (u - u_ext)*(u - u_ext)*dx
E_u = sqrt(assemble(error_u))
print '|| u.e - u.h ||_2 : ', E_u

#error formula ||q-q.h||
error_q = inner(q - q_ext,q - q_ext)*dx
E_q = sqrt(assemble(error_q))
print '|| q.e - q.h ||_2 : ', E_q

#error formula ||lam-lam.h||
error_lam = inner(lam - lam_ext,lam - lam_ext)*dx
E_lam = sqrt(assemble(error_lam))
print '|| lam.e - lam.h ||_2 : ', E_lam
```

Compute the objective function value $J(u_h, q_h)$.

```
#compute ||u.h-u0||^2
ueu0 = inner(u - u0,u - u0)*ds
ue_u0 = (1.0/2)*(assemble(ueu0))
print '1/2|| u.h - u0 ||_O^2 : ', ue_u0



#compute ||q.h||^2
qe = inner(q,q)*dx
q_e = (alpha/2)*(assemble(qe))
print 'a/2|| q.h ||_G^2 : ', q_e


#compute J(u.h,q.h)
print'J(u.h ,q.h) : ',ue_u0+q_e
```

Compute the objective function value $J(u, q)$.

```
#compute ||u-u0||^2
ueu0 = inner(u_ext - u0,u_ext - u0)*ds
ue_u0 = (1.0/2)*(assemble(ueu0))
print '1/2|| u - u0 ||_O^2 : ', ue_u0


#compute ||q||^2
qe = inner(q_ext,q_ext)*dx
q_e = (alpha/2)*(assemble(qe))
print 'a/2|| q ||_G^2 : ', q_e
```

```
#compute J(u,q)
print'J(u ,q) : ',ue_u0+q_e
```

The error analysis formula of objective function corresponding to exact solution, finite element solution.

```
#error objective function
J=ue_u0+q_e
Jh=ue_uh0+qh_e
Je=abs(J - Jh)
print'|J(u,q)-J(uh,qh)|:',Je
```

Convergence rate formulas.

```
#convergence rate u.h
from math import log
for i in range(1, len(E_u)):
r_u = log(E_u[i]/E_u[i-1])/log(h_u[i]/h_u[i-1])
print 'r.u(%.7f) : %.5f'%(h_u[i],r_u)

#convergence rate q.h
from math import log
for j in range(1, len(E_q)):
r_q = log(E_q[j]/E_q[j-1])/log(h_q[j]/h_q[j-1])
print 'r.q(%.7f) : %.5f'%(h_q[j],r_q)

#convergence rate lam.h
from math import log
for k in range(1, len(E_lam)):
r_lam = log(E_lam[k]/E_lam[k-1])/log(h_lam[k]/h_lam[k-1])
print 'r.lam(%.7f) : %.5f'%(h_lam[k],r_lam)

#convergence rate objective function
print'\n'
from math import log
for l in range(1, len(E_J)):
r_J = log(E_J[l]/E_J[l-1])/log(h_J[l]/h_J[l-1])
print 'r.J(%.7f) : %.5f'%(h_J[l],r_J)
```

plot the solution variables and mesh.

```
plot(mesh)
plot(u,title="u")
plot(q,title="q")
plot(lam,title="lam")
interactive()
```

References

[1] S. Larsson, V. Thomée, *Partial differential equations with numerical methods*,
Finite Element Methods for Elliptic Equations (2003).

[2] W. Bangerth, R. Rannacher, *Adaptive finite element methods for differential equations*,
An Abstract Approach for Nonlinear Problems (2003).
Optimization Problems (2003).

[3] Fredi Tröltzsch, *Optimal Control of Partial Differential Equations theory, Methods and Applications*,
Introduction and Examples (2009).

[4] Peter Philip, *Optimal Control of Partial Differential Equations*,
Lecture Notes (November 10, 2009).

[5] A. Logg, Kent A. Mardal , *Automated Solution of Differential Equations by the Finite Element Method*,
The FEniCS Book (2011).