Solving Inverse PDE by the Finite Element Method

Yahya Amad

Supervisor : Stig Larsson

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 control 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

1. **Introduction to Optimal Control** 5  
   1.1. Boundary conditions 5  
   1.2. Dirichlet’s condition 5  
   1.3. Neumann’s condition 5  
   1.4. Robin’s condition 5  
   1.5. Optimal boundary heating 5  
   1.6. Optimal heat source 6  
   1.7. Optimal non-stationary boundary control 6  
   1.8. Optimal vibrations 6  
   1.9. Heating with radiation boundary conditions 7  
   1.10. Simplified superconductivity 7  
   1.11. Control of stationary flows 7  
   1.12. Problems involving semi-linear parabolic equations 7  
   1.13. Control of non-stationary flows 7  
2. **Variational form and Lagrangian function** 8  
   2.1. Variational form of Poisson’s equation 8  
   2.2. Variational form of optimal control problem with Robin’s condition 9  
   2.3. Weak formulation and Lagrangian with Neumann conditions. 9  
   2.4. Variational form and Lagrangian with homogeneous Dirichlet condition 9  
   2.5. Weak form and Lagrangian with Robin’s condition 10  
   2.6. Variational form and Lagrangian with homogeneous constraints 10  
   2.7. Variational form and Lagrangian with inhomogeneous constraints 11  
3. **Finite Element Method** 12  
   3.1. An optimal control problem 12  
   3.2. Construct a variational form 13  
   3.3. Construction of Lagrangian 13  
   3.4. Construct the strong form 14
| 3.5.   | System is indefinite of saddle point form | 15 |
| 3.6.   | Finite element calculation               | 17 |
| 3.7.   | Solve stationary point equations individual by FEM | 17 |
| 3.8.   | An optimal control problem with Neumann boundary condition | 24 |
| 3.9.   | Solve finite element equations individual by FEM | 25 |
| 3.10.  | Solve finite element equations combine by FEM | 27 |
| 3.11.  | Solve finite element equations as one equation by FEM | 31 |
| 3.12.  | Python program                           | 42 |

References 46
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:
\[ u = 0 \text{ on } \partial\Omega \]
A non-homogeneous Dirichlet condition:
\[ u = u_0 \text{ on } \partial\Omega \]

1.3. Neumann’s condition. A homogeneous Neumann condition:
\[ \partial_n u = 0 \text{ on } \partial\Omega \]
A non-homogeneous Neumann condition:
\[ \partial_n u = q \text{ on } \partial\Omega \]

1.4. Robin’s condition. A Robin’s condition is defined as:
\[ \partial_n u = \alpha_0(q - u) \text{ on } \partial\Omega \]

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 \( \Gamma \), calling heat source \( q \), control variable. So the purpose is to find \( q \) such that the temperature distribution \( u \), state variable, in \( \Omega \) is best possible approximation to a required stationary temperature distribution, denoted by \( u_{\Omega} \) or \( u_0 \) in \( \Omega \). It can be describing in mathematical way as follows:
\[
\min J(u,q) = \frac{1}{2} \int_{\Omega} |u - u_{\Omega}|^2 \, dx + \frac{\alpha}{2} \int_{\Gamma} |q|^2 \, ds
\]
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
\[
q_a(x) \leq q(x) \leq q_b(x) \quad \text{on } \Gamma.
\]
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 \( \Gamma \) is denoted by \( n \), \( \alpha \) is heat transfer coefficient from \( \Omega \) 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, \( \Delta \), because otherwise \( \Delta \) 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 $\Omega$. These kinds of problems heated the body $\Omega$ 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:

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

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 $\Omega \subset \mathbb{R}^3$ 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 final 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:

$$\min J(u, q) = \frac{1}{2} \int_{\Omega} \int_{0}^{T} |u(x,T) - u_\Omega(x)|^2 \, dx \, dt + \alpha \frac{1}{2} \int_{\Omega} \int_{0}^{T} |q(x,t)|^2 \, ds \, dt$$

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 $\Omega \subset \mathbb{R}^2$, 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:

$$\min J(u, q) = \frac{1}{2} \int_{\Omega} \int_{0}^{T} |u(x,y) - u_d|^2 \, dx \, dt + \alpha \frac{1}{2} \int_{\Omega} \int_{0}^{T} |q|^2 \, dx \, dt$$

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:

\[ A(u)\phi + B(q, \psi) = 0 \text{ for all } \psi \in V \]

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

\[-\nabla^2 u = f \text{ in } \Omega \]

\[ u = 0 \text{ on } \Gamma \]

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

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

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 \( \Gamma \), 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:

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

where the bilinear form \( a : V \times V \to \mathbb{R} \),

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

To write into abstract form

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

Now, we define the linear and continuous functional \( F : V \to \mathbb{R} \)

\[ F(\varphi) = (f, \varphi)_\Omega \]

Thus, in generalized form

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

\( V^* \) 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

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

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

\[-\int_\Gamma \varphi \partial_n u \, ds + \int_\Omega \nabla u \cdot \nabla \varphi \, dx + \int_\Omega c_0 u \varphi \, dx = \int_\Omega f \varphi \, dx\]

substituting the Robin boundary condition gives

\[-\int_\Gamma \varphi \partial_n u \, ds + \int_\Omega \nabla u \cdot \nabla \varphi \, dx + \int_\Omega c_0 u \varphi \, dx + \int_\Gamma \alpha u \varphi \, ds = \int_\Omega f \varphi \, dx + \int_\Gamma q \varphi \, ds\]

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

\[
\left(\nabla u, \nabla \varphi \right)_\Omega + c_0 \left(u, \varphi \right)_\Omega + \alpha \left(u, \varphi \right)_\Gamma = \left(f, \varphi \right)_\Omega + \left(q, \varphi \right)_\Gamma
\]

for all \(\varphi \in H^1(\Omega)\).

where the linear functional \(L\) is

\[
L(\varphi) = \int_\Omega f \varphi \, dx + \int_\Gamma q \varphi \, ds \text{ for all } \varphi \in H^1(\Omega)
\]

and bilinear form is

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

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

\[
J(u, q) = \frac{1}{2} ||u - u_0||^2_{L_2(\Omega)} + \frac{1}{2} ||q||^2_{L_2(\Gamma_c)}
\]

where \(\Gamma_o\), observational boundary and \(\Gamma_c\), 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

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

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

\[
J(u, q) = \frac{1}{2} ||u - u_0||^2_{L_2(\Omega)} + \frac{\alpha}{2} ||q||^2_{L_\infty(\Omega)}
\]

where \(\Gamma\) is boundary, \(\Omega\) is space domain and \(V = H^1_0(\Omega), Q = L_\infty(\Omega)\)

subjected to the PDE constraint

\[-\nabla^2 u + q u = f \text{ in } \Omega\]
\[u = 0 \text{ on } \Gamma\]
since $qu = u(q)$ chosen $qu = u(q)$ as linear so that $u(q) = q$.

Construct the variational form,

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

Construction of Lagrangian,

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

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

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

construction of variational form

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

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

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

where $\Gamma$ is boundary, $\Omega$ is space domain and $V = H_0^1(\Omega), \, Q = L_\infty(\Omega)$ subjected to the PDE constraint

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

construction of variational form

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

construction of Lagrangian:

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

10
2.7. Variational form and Lagrangian with inhomogeneous constraints. minizm the optimal control problem

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

where \( \Gamma \) is boundary, \( \Omega \) 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

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

construction of Lagrangian:

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

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

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

\[
L'_u(u, q, \lambda)(\varphi) = J'_u(u, q)(\varphi) - A'(u)(\varphi, \lambda) = 0 \text{ for all } \varphi \in V
\]
\[
L'_q(u, q, \lambda)(\chi) = J'_q(u, q)(\varphi) - B'(q)(\chi, \lambda) = 0 \text{ for all } \chi \in Q
\]
\[
L'_\lambda(u, q, \lambda)(\psi) = -A(u)(\psi) - B(\psi, q) = 0 \text{ for all } \psi \in V
\]

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

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

3.1. An optimal control problem. Consider a body with domain \( \Omega \subset \mathbb{R}^2 \) which want to be heated or cooled. So that enter the heat source \( q \) at boundary \( \Gamma_c \), 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 \( \Gamma_o \), observational boundary. The optimal function \( J(u, q) \) measures the derivation of the solution \( u \) of dirichlet values along the boundary \( \Gamma_o \), observation boundary, from the prescribed function \( u_0 \).

To minimise the objective function

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

where \( \Gamma_o \), observational boundary and \( \Gamma_c \), 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

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

where \(ds\) denotes surface element and \(n\) denotes the outward unit normal to \(\Gamma_o\) and \(\Gamma_c\), the heat transmission coefficient from \(\Omega\) to surrounding medium is represented by function \(\alpha\), 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 \(\Delta\) is not coercive operator.

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

3.2. Construct a variational form.

\[
-\int_{\Omega} \nabla^2 u \varphi \, dx + \int_{\Omega} s(u)\varphi \, dx = \int_{\Omega} f\varphi \, dx \\
-\int_{\Omega} \nabla \cdot u \varphi \, dx + \int_{\Omega} s(u)\varphi \, dx = \int_{\Omega} f\varphi \, dx \\
-\int_{\Omega} \varphi \nabla u \cdot n \, ds + \int_{\Omega} \nabla \cdot \varphi \, dx + \int_{\Omega} s(u)\varphi \, dx = \int_{\Omega} f\varphi \, dx \\
-\int_{\Gamma_c} \varphi \partial_n u \, ds + \int_{\Omega} \nabla u \cdot \varphi \, dx + \int_{\Omega} s(u)\varphi \, dx = \int_{\Omega} f\varphi \, dx \\
-\int_{\Gamma_c} \varphi q \, ds + \int_{\Omega} \nabla u \cdot \varphi \, dx + \int_{\Omega} s(u)\varphi \, dx = \int_{\Omega} f\varphi \, dx \\
(q, \varphi)_{\Gamma_c} - (\nabla u, \nabla \varphi)_{\Gamma_c} - (s(u), \varphi)_{\Gamma_c} = -(f, \varphi)_{\Gamma_c} \\
(\nabla u, \nabla \varphi)_{\Gamma_o} + (s(u), \varphi)_{\Gamma_o} - (f, \varphi)_{\Gamma_o} = 0 \text{ for all } \varphi \in V = H^1
\]
since \(\varphi\) is well–defined by trace theorem: for bounded domain \(\Omega\) with smooth boundary \(\Gamma\), map \(C^1(\Omega) \to C(\Gamma_c) \text{ may be extended to } H^1 \to L_2(\Gamma_c) \text{ with } |||\varphi|||_{L_2(\Gamma)} \leq C|||\varphi|||_1 \text{ for all } \varphi \in H^1(\Omega) = V \)

\[A(u)(\varphi) = (\nabla u, \nabla \varphi)_{\Gamma_o} + (s(u), \varphi)_{\Gamma_o} - (f, \varphi)_{\Gamma_o} \text{ is semi-linear in } \varphi \]
\[B(q, \varphi) = -(q, \varphi)_{\Gamma_c} \text{ is linear.} \]

3.3. Construction of Lagrangian.

\[L(u, q, \lambda) = J(u, q) - A(u)(\lambda) - B(q, \lambda)\]

\[L(u, q, \lambda) = \frac{1}{2}||u - u_0||_{\Gamma_o}^2 + \frac{1}{2}||q||_{\Gamma_c}^2 - (\nabla u, \nabla \lambda)_{\Omega} - (s(u), \lambda)_{\Omega} + (f, \lambda)_{\Omega} + (q, \lambda)_{\Gamma_c}
\]

where

\[J'_u(u, q)(\varphi) = (\varphi, u - u_0)_{\Gamma_o}\]
\[J'_u(u, q)(\varphi) = \alpha(\varphi, q)_{\Gamma_c}\]
\[A'(u)(\varphi, \lambda) = (\nabla \varphi, \nabla \lambda) + (s'(u)\varphi, \lambda)\]
\[B'(q)(\chi, \lambda) = -(\chi, \lambda)_{\Gamma_c} = B(\chi, \lambda)\]

so that

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

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

1. for all \( \varphi \in H_0^1 : (\varphi, u - u_0)_{\Gamma_e} - (\varphi, \nabla \lambda \cdot n)_{\Gamma} + (\varphi, \nabla^2 \lambda - s'(u) \lambda)_{\Omega} = 0 \)

Since \( \varphi \in H_0^1 \) therefore \( \varphi = 0 \) on \( \Gamma_o, \Gamma \)
\( \nabla^2 \lambda - s'(u) \lambda = 0 \) in \( \Omega \)
2. for all \( \varphi \in H^1 : (\varphi, u - u_0 - \nabla \lambda \cdot n)_{\Gamma_e} - (\varphi, \nabla \lambda \cdot n)_{\Gamma_e} = 0 \)

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 \( \Gamma_o \)
\( u - u_0 = \partial_n \lambda \) on \( \Gamma_o \)

The strong form of \( \lambda \) equation

\[
-\nabla^2 \lambda + \lambda = 0 \quad \text{in } \Omega \\
\partial_n \lambda = 0 \quad \text{on } \partial \Omega \setminus \Gamma_o \\
u - u_0 = \partial_n \lambda \quad \text{on } \Gamma_o
\]

\[
L'_u(u, q, \lambda)(\chi) = \alpha(\chi, q)_{\Gamma_e} + (\chi, \lambda)_{\Gamma_e} = (\chi, \alpha q + \lambda)_{\Gamma_e} = 0 \quad \text{for all } \chi \in L^2(\Gamma_e)
\]

The strong form of \( q \) equation

\[
\alpha q + \lambda = 0 \quad \text{on } \Gamma_c
\]

\[
L'_u(u, q, \lambda)(\psi) = -(\nabla u, \nabla \psi)_{\Omega} - (s(u), \psi)_{\Omega} + (q, \psi)_{\Gamma_e} + (f, \varphi) = 0 \\
(\nabla^2 u, \psi)_{\Omega} - (\nabla u \cdot n, \psi)_{\Gamma} - (s(u), \psi)_{\Omega} + (q, \psi)_{\Gamma_e} + (f, \varphi) = 0 \\
(\nabla^2 u, \psi)_{\Omega} - (\nabla u \cdot n, \psi)_{\Gamma_e} - (\nabla u \cdot n, \psi)_{\Gamma_e} - (s(u), \psi)_{\Omega} + (q, \psi)_{\Gamma_e} + (f, \varphi) = 0 \\
(\nabla^2 u - s(u) + f, \psi)_{\Omega} + (-\nabla u \cdot n + q, \psi)_{\Gamma_e} = 0 \quad \text{for all } \psi \in H^1
\]

1. for all \( \psi \in H_0^1 : (\nabla u \cdot n, \psi)_{\Gamma_e} + (\nabla^2 u - s(u) + f, \psi)_{\Omega} + (-\nabla u \cdot n + q, \psi)_{\Gamma_e} = 0 \\
\nabla^2 u - s(u) + f = 0 \quad \text{in } \Omega \\
-\nabla^2 u + s(u) = f \quad \text{in } \Omega \\
2. for all \( \psi \in H^1 : (\nabla u \cdot n + q, \psi)_{\Gamma_e} = 0 \\
3. \psi = 0 \quad \text{on } \Gamma_c : (\nabla u \cdot n, \psi)_{\partial \Omega \setminus \Gamma_e} = 0 \\
-\partial_n u = -\nabla u = 0 \quad \text{on } \partial \Omega \setminus \Gamma_c \\
4. for all \( \psi \in H^1 : (-\nabla u \cdot n + q, \psi)_{\Gamma_e} = 0 \\
-\nabla u \cdot n + q = 0 \quad \text{on } \Gamma_c \\
\partial_n u = \nabla u \cdot n = q \quad \text{on } \Gamma_c \\
\]

The strong form of \( u \) equation

\[
-\nabla^2 u + s(u) = f \quad \text{in } \Omega \\
\partial_n u = q \quad \text{on } \Gamma_c \\
\partial_n u = 0 \quad \text{on } \partial \Omega \setminus \Gamma_c
\]
describe the stationary point equations as finite element equations.

\[ L_\nu'(u_h, q_h, \lambda_h)(\varphi) = \rho_\nu'(u_h, q_h, \lambda_h)(\varphi_h) = (\varphi_h, u_h - u_0)_{\Gamma_n} - (\nabla \varphi_h, \nabla \lambda_h)_{\Omega} - (s'(u_h)\varphi_h, \lambda_h)_{\Omega} = 0 \]
\[ \forall \varphi_h \in H^1 \]
\[ L_\nu'(u_h, q_h, \lambda_h)(\chi) = \rho_\nu'(u_h, q_h, \lambda_h)(\chi_h) = \alpha(\chi_h, q_h)_{\Gamma_n} + (\chi_h, \lambda_h)_{\Gamma_n} = 0 \]
\[ \forall \chi_h \in L_2(\Gamma_c) \]

\[ L_\nu'(u_h, q_h, \lambda_h)(\psi_h) = \rho_\nu'(u_h, q_h, \lambda_h)(\psi_h) = -(\nabla u_h, \nabla \psi_h)_{\Omega} - (s(u_h), \psi_h)_{\Omega} + (q_h, \psi_h)_{\Gamma_n} + (f, \psi_h) \]
\[ \forall \psi_h \in H^1 \]

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 \( \Omega \) 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

\[ L_\nu'(u, q, \lambda)(\varphi) = (\varphi, u - u_0)_{\Gamma_n} - (\nabla \varphi, \nabla \lambda)_{\Omega} - (s'(u)\varphi, \lambda)_{\Omega} = 0 \]
\[ (\varphi, u)_{\Gamma_n} - (\varphi, u_0)_{\Gamma_n} - (\nabla \varphi, \nabla \lambda)_{\Omega} - (s(u)\varphi, \lambda)_{\Omega} = 0 \]
\[ (\Phi_j, \sum_{i=1}^n u_i \Phi_i(x))_{\Gamma_n} - (\Phi_j, \sum_{i=1}^n u_0 \Phi_i(x))_{\Gamma_n} - (\nabla \Phi_j, \nabla \sum_{i=1}^n \lambda_i \Phi_i(x))_{\Omega} - (\Phi_j, \sum_{i=1}^n \lambda_i \Phi_i(x))_{\Omega} = 0 \]
\[ (\Phi_j, \sum_{i=1}^n (u_i - u_0) \Phi_i(x))_{\Gamma_n} - \sum_{i=1}^n \lambda_i (\nabla \Phi_j, \nabla \Phi_i(x))_{\Omega} - \sum_{i=1}^n \lambda_i (\Phi_j, \Phi_i(x))_{\Omega} = 0 \]

\[ \lambda \text{ equation into the matrix form} \]

\[ \begin{bmatrix} [u_{11} - u_{10}u_2 - u_{20} \cdots u_n - u_{n0}] \end{bmatrix} \begin{bmatrix} (\Phi_j, \Phi_1)_{\Gamma_n} \\ (\Phi_j, \Phi_2)_{\Gamma_n} \\ \vdots \\ (\Phi_j, \Phi_n)_{\Gamma_n} \end{bmatrix} = \begin{bmatrix} (\nabla \Phi_j, \nabla \Phi_1)_{\Omega} \\ (\nabla \Phi_j, \nabla \Phi_2)_{\Omega} \\ \vdots \\ (\nabla \Phi_j, \nabla \Phi_n)_{\Omega} \end{bmatrix} \begin{bmatrix} \lambda_1 \lambda_2 \cdots \lambda_n \end{bmatrix} \]
\[ = 0 \]

or

\[ (\tilde{u} - \tilde{u}_0)^T A_{\Gamma_n} - \lambda^T B_{\Omega} - \lambda^T A_{\Omega} = 0 \]
\[ A_{\Gamma_n}^T (\tilde{u} - \tilde{u}_0) - B_{\Omega}^T \lambda - A_{\Omega}^T \lambda = 0 \]
\[ 15 \text{ ....... (1)} \]
Write the following equations:

\( L_{\eta}(u, q, \lambda)(\chi) = \alpha(\chi, q)u_{\eta} + (\chi, \lambda)u_{\eta} = (\chi, \alpha q + \lambda)u_{\eta} = 0 \)

\( (\chi, \alpha q + \lambda)u_{\eta} = 0 \) for all \( \chi \in L_{2}(\Gamma_{E}) \)

\( \sum_{i=1}^{n} \Phi_{j}(\alpha q_{i} + \lambda_{i})\Phi_{i}(x) = \sum_{i=1}^{n} \Phi_{j}(\alpha q_{i} + \lambda_{i})\Phi_{i}(x) = 0 \)

\( n \)

\( q \) equation into the matrix form

\[ \begin{bmatrix} \alpha_{1} & \lambda_{1} & \alpha_{2} & \lambda_{2} & \cdots & \alpha_{n} & \lambda_{n} \end{bmatrix} \]

or

\[ -\alpha q^{T}A_{\Gamma_{E}} = \tilde{\lambda}^{T}A_{\Gamma_{E}} \]

\[ \alpha \tilde{\lambda}^{T} \tilde{q} + A_{\Gamma_{E}}^{T} \tilde{\lambda} = 0 \] ........ (2)

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

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

\( -\sum_{i=1}^{n} u_{i} \Phi_{i}(x) \nabla \Phi_{j} = \sum_{i=1}^{n} u_{i} \Phi_{i}(x) \Phi_{j} + \sum_{i=1}^{n} q_{i} \Phi_{i}(x) \Phi_{j} + \sum_{i=1}^{n} f_{i} \Phi_{i}(x) \Phi_{j} = 0 \)

\( -\sum_{i=1}^{n} u_{i} \Phi_{i}(x) \Phi_{j} = \sum_{i=1}^{n} u_{i} \Phi_{i}(x) \Phi_{j} + \sum_{i=1}^{n} q_{i} \Phi_{i}(x) \Phi_{j} + \sum_{i=1}^{n} f_{i} \Phi_{i}(x) \Phi_{j} = 0 \)

\( u \) equation into the matrix form

\[ \begin{bmatrix} \alpha(\Phi_{j}, \Phi_{1})_{\Omega} \\ \alpha(\Phi_{j}, \Phi_{2})_{\Omega} \\ \vdots \\ \alpha(\Phi_{j}, \Phi_{n})_{\Omega} \end{bmatrix} + \begin{bmatrix} -u_{1} + f_{1} \\ -u_{2} + f_{2} \\ \vdots \\ -u_{n} + f_{n} \end{bmatrix} = \begin{bmatrix} \alpha(\Phi_{j}, \phi_{1})_{\Omega} \\ \alpha(\Phi_{j}, \phi_{2})_{\Omega} \\ \vdots \\ \alpha(\Phi_{j}, \phi_{n})_{\Omega} \end{bmatrix} \]

\[ + \begin{bmatrix} (\Phi_{j}, \phi_{1})_{\Omega} \\ (\Phi_{j}, \phi_{2})_{\Omega} \\ \vdots \\ (\Phi_{j}, \phi_{n})_{\Omega} \end{bmatrix} = 0 \]

\( -\tilde{u}^{T}B_{\Omega} + (-\tilde{u} + \tilde{f})^{T}A_{\Omega} + q^{T}A_{\Gamma_{E}} = 0 \)

\( -(B_{\Omega} + A_{\Omega})\tilde{u}^{T} + A_{\Gamma_{E}}\tilde{q}^{T} = -f^{T}A_{\Omega} \) ........ (3)

Write the following equations:

\[ A_{\Gamma_{E}}^{T}u_{\eta} - (B_{\Omega}^{T} + A_{\Omega}^{T})\tilde{\lambda} = A_{\Gamma_{E}}^{T}u_{\eta} \]

\[ \alpha A_{\Gamma_{E}}^{T}\tilde{q} + A_{\Gamma_{E}}^{T}\tilde{\lambda} = 0 \]

\( -(B_{\Omega} + A_{\Omega})\tilde{u} + A_{\Gamma_{E}}\tilde{q} = -A_{\Omega}f \)
Minimise the objective function subject to PDE’s constraints. To solve finite element equations individually use this case as a test example when

\[ q = 1 \]

and

\[ f = \frac{y^2}{2} - y + \frac{1}{17} \]

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 \( \Gamma_c \) and \( f = \frac{y^2}{2} - y + \frac{1}{17} \). 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.

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

3.7. Solve stationary point equations individually 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 use this case as an test example. Minimise the objective function

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

where \( \Gamma_o \), observational boundary and \( \Gamma_c \), control boundary subjected to constraints

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

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.

\[ A^T \begin{bmatrix} \alpha A_{f,c} & -B_{f,c} + A_{c} \\ 0 & \alpha A_{c} \end{bmatrix} \begin{bmatrix} u \\ q \end{bmatrix} = \begin{bmatrix} A^T_{e}u_0 \\ 0 \end{bmatrix} \]

the block form of the matrix is:

\[ \begin{bmatrix} A_c & B_e^T \\ B_e & 0 \end{bmatrix} \]

where

\[ A_c = \begin{bmatrix} A_{f,c}^T & 0 \\ 0 & A_{c} \end{bmatrix}, \quad B_c = \begin{bmatrix} -(B_{f,c} + A_{c}) \\ A_{c} \end{bmatrix} \]

\[ B_e^T = \begin{bmatrix} -(B_{f,c} + A_{c}) \\ A_{c} \end{bmatrix}, \quad b_c = \begin{bmatrix} A^T_{c}u_0 \\ -fA_{c} \end{bmatrix} \]

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

\[ \lambda \text{ equation} \]

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

\[ q \text{ equation} \]

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

\[ u \text{ equation} \]

\[ L'_u(u, q, \lambda)(\psi) = -(\nabla u, \nabla \psi)_{\Omega} - (u, \psi)_{\Omega} + (q, \psi)_{\Gamma_c} + (f, \psi) = 0 \text{ for all } \psi \in H^1 \]
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 $\Gamma_c$ and $\Gamma_o$ respectively. Thus the maximum value of $u$ lies on the boundary $\Gamma_c$ where $y = 0$ and the minimum value of $u$ lies on the boundary $\Gamma_o$ 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.

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

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

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**

| $h$   | $||u - u_h||_{L^2}$ |
|-------|---------------------|
| 1/4   | 0.00602560985088000 |
| 1/8   | 0.00151851498406000 |
| 1/16  | 0.00038058129942700 |
| 1/32  | 0.00009521763865940 |
| 1/64  | 0.00002380988317370 |
| 1/128 | 0.00000595533940000 |

**Table 2. Convergence rate**

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.98845</td>
</tr>
<tr>
<td>1/8</td>
<td>1.99638</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99890</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99967</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99978</td>
</tr>
</tbody>
</table>

**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 $V$.

$$V = \text{FunctionSpace}(\text{mesh}, \text{"Lagrange"}, 1)$$

Specify the boundary parts $\Gamma_o$, $\Gamma_c$, and $\Gamma \setminus \Gamma_o$.

```python
boundary-parts = \text{MeshFunction}(\text{"uint"}, \text{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 $\Gamma_c$ is marked by 0, the observational boundary $\Gamma_o$ is marked by 1.

```python
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, $\Gamma \setminus \Gamma_o$ is defined as:

```python
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 $f$.

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

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

```python
a = \text{inner}(\text{grad}(u), \text{grad}(v))*dx + \text{inner}(u,v)*dx
L = f*v*ds + q*v*ds(0)
```

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

```python
A = \text{assemble}(a, \text{exterior_facet_domains}=\text{boundary-parts})
b = \text{assemble}(L, \text{exterior_facet_domains}=\text{boundary-parts})
print'\text{A : }', A.array()
print'\text{b : }', b.array()
```
To define solution variable.

\[ u = \text{Function}(V) \]
\[ U = u.\text{vector()} \]
\[ \text{solve}(A, U, b) \]

Define error norm in space \( L_2 \).

\[ \text{error}_2 = \text{inner}(u - u_{\text{ext}}, u - u_{\text{ext}})\*dx \]
\[ E = \sqrt{\text{assemble}(\text{error}_2)} \]

Choose \( u - u_0 \) and solve for \( \lambda \) in strong form for exact solution.

Now task is to solve the variational equation for \( \lambda \) keeping \( u - u_0 \) known. Previously solved for \( u \) keeping \( q \) known. For exact solution, chosen \( u - u_0 = 1.0 \) on \( \Gamma_o \), so that the strong form give the \( \lambda \)–value by using boundary conditions. The \( \lambda \)–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 \textit{hyperbolic cosine} and \textit{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 \( \lambda \) 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 \( \Gamma_c \) which is minimum value of \( \lambda \) and \( \lambda = \frac{\cosh(1)}{\sinh(1)} = 1.3130352854993312 \) on \( \Gamma_o \) which is maximum value of \( \lambda \) over space \( \Omega \).

Thus the maximum value of \( \lambda \) is occurred on the boundary \( \Gamma_o \) where \( y = 1 \) and the minimum value of \( \lambda \) is occurred on the boundary \( \Gamma_c \) where \( y = 0 \).

Choose \( u - u_0 \) and solve for \( \lambda_h \) in variational form finite element solution.

Now the aim is to calculate \( \lambda_h \)-value corresponding to various meshes by \textit{finite element method}. The method which used as an approximate solution is \textit{finite element method} which is computed in python. Now, chosen \( u - u_0 = 1.0 \) and compute \( \lambda_h \) - 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 \( \lambda \)–value thus the \( \lambda \) 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 \( \lambda \)-values when \( h = \frac{1}{4} \) is \( \lambda_h = 0.84745815435 \) 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      | 0.003091941257572000                |
| 1/8      | 0.0007906375867200                  |
| 1/16     | 0.0002016675826143                  |
| 1/32     | 0.0000505849827850                  |
| 1/64     | 0.00001265907791610                 |
| 1/128    | 0.000002977557572077                |
### Table 4. Convergence rate

<table>
<thead>
<tr>
<th>h</th>
<th>$r_{h}L_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.953414</td>
</tr>
<tr>
<td>1/8</td>
<td>1.985000</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99528</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99854</td>
</tr>
<tr>
<td>1/64</td>
<td>2.08796</td>
</tr>
</tbody>
</table>

**Program syntax for $\lambda$ in Python**

Space and boundary parts are defined same as before.
Test and trial functions are defined.

```python
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.

```python
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.

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

The solution function is defined.

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

The error norm is defined in $L^2$ space.

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

The convergence rate is defined.

```python
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 $\Gamma_c$ 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 $h = 1/4$ 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 $\Gamma_c$ and $\Gamma_o$ respectively. Thus the maximum value of $u$ lies on the boundary $\Gamma_c$ where $y = 0$ and the minimum value of $u$ lies on the boundary $\Gamma_o$ 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 $h = 1/4$ 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.

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

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

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 $1/4$ as step size $h$ is decreasing.

Table 5. Error Analysis

| $h$ | $||u - u_h||_{L^2}$ |
|-----|---------------------|
| 1/4 | 0.0030914129601800 |
| 1/8 | 0.0007983675978610 |
| 1/16| 0.0002016782637620 |
| 1/32| 0.0000505850975933 |
| 1/64| 0.0000126596551221 |
| 1/128| 0.0000031687821931 |

Table 6. Convergence rate

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_{u_{L^2}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.95314</td>
</tr>
<tr>
<td>1/8</td>
<td>1.98500</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99627</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99847</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99824</td>
</tr>
</tbody>
</table>

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 $\Gamma_c$ 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.

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

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 \( u = 2.61119902902 \) and \( u = 2.22484755724 \) when \( y = 0 \) and \( y = 1 \) respectively. These are the boundary parts thus maximum of \( u \) occurred on the boundary \( \Gamma_c \) where \( y = 0 \) and minimum of \( u \) occurred on \( \Gamma_o \) 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 \( u_h = 2.62551088775 \) and \( u_h = 2.2555860553 \) 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 \( \text{assemble} \) which is integrated over space.

\[
||u - u_h||_{L^2} = \left( \int |u - u_h|^2 \, dx \right)^{1/2}
\]

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

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      | 0.000694129979160               |
| 1/32      | 0.0001735807512930               |
| 1/64      | 0.000043012460430               |
| 1/128     | 0.000010204318957               |

**Table 8. Convergence rate**

<table>
<thead>
<tr>
<th>( h )</th>
<th>( r_{u_{L^2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.99365</td>
</tr>
<tr>
<td>1/8</td>
<td>1.99793</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99936</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99980</td>
</tr>
<tr>
<td>1/64</td>
<td>2.08855</td>
</tr>
</tbody>
</table>
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:

\[
J(u, q) = \frac{1}{2}||u - u_0||_{\Gamma_0}^2 + \frac{1}{2}\alpha||q||_{\Omega}^2
\]

where \(\Gamma_0\), observational boundary, subjected to constraints

\[-\nabla^2 u + u = q \text{ in } \Omega \]
\[\partial_n u = 0 \text{ on } \Gamma\]

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:

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

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

\[
L'_u(u, q, \lambda)(\varphi) = (\varphi, u - u_0)_{\Gamma_0} - (\nabla \varphi, \nabla \lambda)_{\Omega} - (\varphi, \lambda)_{\Omega} = 0 \text{ for all } \varphi \in H^1(\Omega)
\]
\[
L'_q(u, q, \lambda)(\chi) = \alpha(\chi, q)_{\Omega} + (\chi, \lambda)_{\Omega} = 0 \text{ for all } \chi \in L_2(\Omega)
\]
\[
L'_\lambda(u, q, \lambda)(\psi) = -(\nabla u, \nabla \psi)_{\Omega} - (u, \psi)_{\Omega} + (q, \psi)_{\Omega} = 0 \text{ for all } \psi \in H^1(\Omega)
\]

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

\[
(\varphi, u - u_0)_{\Gamma_0} - (\nabla \varphi, \nabla \lambda)_{\Omega} - (\varphi, \lambda)_{\Omega} = 0 \text{ for all } \varphi \in H^1(\Omega)
\]

\[
\begin{align*}
-\nabla^2 \lambda + \lambda &= 0 \text{ in } \Omega \\
\partial_n \lambda &= u - u_0 \text{ on } \Gamma_0 \\
\partial_n \lambda &= 0 \text{ on } \Gamma \setminus \Gamma_0
\end{align*}
\]

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

\[
\alpha q + \lambda = 0 \text{ in } \Omega
\]

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

\[
\begin{align*}
-\nabla^2 u + u &= q \text{ in } \Omega \\
\partial_n u &= 0 \text{ on } \Gamma
\end{align*}
\]

Since \(\alpha\) and \(u_0\) are involved in the stationary point equations therefore the solution term contains \(\alpha\) and \(u_0\).
The exact solution is:

\[
\begin{align*}
  u &= \frac{A}{\alpha} \left[ \frac{-\cosh(y)}{2 \sinh(1)} (1 + \coth(1)) + \frac{y \sinh(y)}{2 \sinh(1)} \right] = \frac{A}{\alpha} v(y) \\
  q &= -\frac{A \cosh(y)}{\alpha \sinh(1)} \\
  \lambda &= \frac{A \cosh(y)}{\sinh(1)} \\

\text{where } A &= u \big|_{\Gamma_0} - u_0 = \frac{A}{\alpha} v(1) - u_0 \\
\text{implies } u_0 &= A \left( \frac{v(1) - \alpha}{v(1) - \alpha} \right) \\
\text{so that } A &= \frac{u_0}{v(1) - \alpha} \text{ or } \frac{A}{\alpha} = \frac{u_0}{v(1) - \alpha}
\end{align*}
\]

Thus

\[
\begin{align*}
  u &= \frac{u_0}{v(1) - \alpha} v(y) \\
  q &= \frac{-u_0 \cosh(y)}{(v(1) - \alpha) \sinh(1)} \\
  \lambda &= \frac{-u_0 \cosh(y)}{(v(1) - \alpha) \sinh(1)} \\

\text{The solution behaviour when } \alpha \to 0
\end{align*}
\]

Thus

\[
\begin{align*}
  u &= \frac{u_0}{v(1) - \alpha} v(y) \to \frac{u_0}{v(1)} v(y) \text{ in } \Omega \text{ when } \alpha = 0 \\
  \text{and } u &= \frac{u_0}{v(1) - \alpha} v(y) \to u_0 \text{ on } \Gamma_0 \text{ when } \alpha = 0 \\
  q &= \frac{-u_0 \cosh(y)}{(v(1) - \alpha) \sinh(1)} \to \frac{-u_0 \cosh(y)}{v(1) \sinh(1)} \text{ when } \alpha = 0 \\
  \lambda &= \frac{-u_0 \cosh(y)}{(v(1) - \alpha) \sinh(1)} \to 0 \text{ when } \alpha = 0 \\
  u \big|_{\Gamma_0} - u_0 &= u_0 \frac{v(1)}{v(1) - \alpha} - u_0 = \frac{u_0}{v(1) - \alpha} \frac{v(1) - v(1) + \alpha}{v(1) - \alpha} = \frac{\alpha u_0}{v(1) - \alpha} \\
  \text{where } u \big|_{\Gamma_0} &= u_0 \frac{v(1)}{v(1) - \alpha} = u_0 \frac{v(1) - \alpha + \alpha}{v(1) - \alpha} = u_0 (1 + \frac{\alpha}{v(1) - \alpha}) \to u_0 \text{ when } \alpha = 0
\end{align*}
\]

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 \( u_0 = 1.0, q = \frac{\cosh(y)}{v(1) - \alpha} \) and 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:

\[
||u - u_h||_{L^2} = \left( \int |u - u_h|^2 \, dx \right)^{1/2}
\]

\[
r_u = \frac{\ln(E_i/E_{i+1})}{\ln(h_i/h_{i+1})}
\]

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 1/4 as step size \( h \) is decreasing.
Choose $u_0$, $u$ and solve for $\lambda_h$ by finite element method.

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

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

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

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.
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 \( \lambda \)-equation then the stationary point equations become:

\[
(\varphi, u)_\Gamma - (\nabla \varphi, \nabla \lambda)_\Omega - (\varphi, \lambda)_\Omega = (\varphi, u_0)_\Gamma \quad \text{for all } \varphi \in H^1(\Omega)
\]

\[
-(\nabla u, \nabla \psi)_\Omega - (u, \psi)_\Omega - \frac{1}{\alpha}(\lambda, \psi)_\Omega = 0 \quad \text{for all } \psi \in H^1(\Omega)
\]

solve these equations for \( u, \lambda \) keeping \( u_0 \) known for various \( \alpha \)'s and compute error, convergence rate.

Equations in matrix form

\[
\begin{bmatrix}
A^T_{\Gamma,\Gamma} & -\left((B^T_{\Omega} + A^T_{\Omega})\right) \\
-(B_{\Omega} + A_{\Omega}) & \frac{1}{\alpha}A_{\Omega}
\end{bmatrix}
\begin{bmatrix}
\bar{u} \\
\bar{\lambda}
\end{bmatrix}
=
\begin{bmatrix}
A^T_{\Gamma,\Gamma} u_0 \\
0
\end{bmatrix}
\]

into matrix form as follows:

\[
\begin{bmatrix}
A^T_{\Gamma,\Gamma} & -\left((B^T_{\Omega} + A^T_{\Omega})\right) \\
-(B_{\Omega} + A_{\Omega}) & \frac{1}{\alpha}A_{\Omega}
\end{bmatrix}
\begin{bmatrix}
[u] \\
[\lambda]
\end{bmatrix}
=
\begin{bmatrix}
[A^T_{\Gamma,\Gamma} u_0] \\
0
\end{bmatrix}
\]

where the square matrix has eigenvalues

\[
\frac{A_{\Gamma,\Gamma} - \frac{1}{\alpha} A_{\Omega}}{2} \pm \sqrt{\frac{(A_{\Gamma,\Gamma} - \frac{1}{\alpha} A_{\Omega})^2}{4} + (B_{\Omega} + A_{\Omega})(B^T_{\Omega} + A^T_{\Omega}) + \frac{1}{\alpha} A_{\Omega} A_{\Gamma,\Gamma}},
\]

one is positive, one is negative, therefore it is a saddle point.
The numerical form of the matrices when $\alpha = 1$ with mesh $(2, 1)$:

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

$$
-(B_{\Omega} + A_{\Omega}) = \begin{bmatrix}
-1.3 & 0.98 & 0 & 0.23 & -0.042 & 0 \\
0.98 & -2.6 & 0 & 0.458 & -0.042 & 0 \\
0.98 & -1.3 & 0 & 0 & 0 & 0.23 \\
0.23 & 0 & 0 & -1.3 & 0.98 & 0 \\
-0.042 & 0.458 & 0 & 0.98 & -2.6 & 0.98 \\
0 & -0.042 & 0 & 0.23 & 0 & 0.98 & -1.3
\end{bmatrix},
$$

$$
-(B_{\Omega}^T + A_{\Omega}^T) = \begin{bmatrix}
-1.3 & 0.98 & 0 & 0.23 & -0.042 & 0 \\
0.98 & -2.6 & 0 & 0.458 & -0.042 & 0 \\
0.98 & -1.3 & 0 & 0 & 0 & 0.23 \\
0.23 & 0 & 0 & -1.3 & 0.98 & 0 \\
-0.042 & 0.458 & 0 & 0.98 & -2.6 & 0.98 \\
0 & -0.042 & 0 & 0.23 & 0 & 0.98 & -1.3
\end{bmatrix},
$$

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

$$
\frac{-1}{\alpha}A_{\Omega} = \begin{bmatrix}
.083 & -0.0208 & 0 & -0.0208 & -0.042 & 0 \\
-0.0208 & -0.125 & -0.0208 & 0 & -0.042 & -0.042 \\
0 & -0.0208 & -0.042 & 0 & 0 & -0.0208 \\
-0.0208 & 0 & 0 & -0.042 & -0.0208 & 0 \\
-0.042 & -0.042 & 0 & -0.0208 & -0.125 & -0.0208 \\
0 & -0.042 & -0.0208 & 0 & -0.0208 & -0.083
\end{bmatrix}
$$

the matrix form

$$
\begin{bmatrix}
A_{T_{v}u_0}^T & -(B_{\Omega}^T + A_{\Omega}^T) \\
-(B_{\Omega} + A_{\Omega}) & \frac{-1}{\alpha}A_{\Omega}
\end{bmatrix} =
\begin{bmatrix}
.5 & .083 & 0 & .167 & 0 & 0 & -1.3 & .98 & 0 & 0.23 & -0.042 & 0 \\
.083 & .33 & .083 & 0 & 0 & 0 & .98 & -2.6 & .98 & 0 & 0.458 & -0.042 \\
.083 & .33 & .083 & 0 & 0 & 0 & .98 & -1.3 & 0 & 0 & 0 & .23 \\
.167 & 0 & 0 & .33 & 0 & 0 & .23 & 0 & 0 & -1.3 & 0.98 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.042 & 0.458 & 0 & .98 & -2.6 & .98 \\
0 & 0 & .167 & 0 & 0 & 0 & .33 & 0 & -0.042 & .23 & 0 & 0.98 & -1.3 \\
-1.3 & .98 & 0 & .23 & -0.042 & 0 & .083 & -0.0208 & 0 & -0.0208 & -0.042 & 0 \\
.98 & -2.6 & .98 & 0 & 0.458 & -0.042 & -0.0208 & -1.25 & -0.0208 & 0 & -0.042 & -0.042 \\
.98 & -1.3 & 0 & 0 & 23 & 0 & -0.0208 & -0.042 & 0 & 0 & -0.0208 \\
.23 & 0 & 0 & -1.3 & .98 & 0 & -0.0208 & 0 & 0 & -0.042 & -0.0208 & 0 \\
-0.042 & 0.458 & 0 & .98 & -2.6 & .98 & -0.042 & -0.042 & 0 & -0.0208 & -1.25 & -0.0208 \\
0 & -0.042 & 0.23 & 0 & 0.98 & -1.3 & 0 & -0.042 & -0.0208 & 0 & -0.0208 & -0.083
\end{bmatrix}
$$
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   | 0.000127187161463000 | 0.002802482408900000           |
| 1/8   | 0.000339252793065000 | 0.0007060069555695000           |
| 1/16  | 0.000008646124403300 | 0.000176929218032000           |
| 1/32  | 0.000002174207966880 | 0.000042655006885500           |
| 1/64  | 0.000000541205787071 | 0.000011068859373600           |
| 1/128 | 0.000000138049496300 | 0.000002767334468500           |
| 1/256 | 0.000000027878155806 | 0.000000691581891606           |

Table 14. Convergence rate

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_{\lambda}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.90652</td>
<td>1.98895</td>
</tr>
<tr>
<td>1/8</td>
<td>1.97224</td>
<td>1.99651</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99156</td>
<td>1.99994</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99827</td>
<td>1.99968</td>
</tr>
<tr>
<td>1/64</td>
<td>2.01328</td>
<td>2.00053</td>
</tr>
<tr>
<td>1/128</td>
<td>2.27367</td>
<td>2.00053</td>
</tr>
</tbody>
</table>

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   | 0.000253103599223300 | 0.000515429710174000           |
| 1/8   | 0.000679699797698000 | 0.001304997868500000           |
| 1/16  | 0.001738093002070000 | 0.003273429784100000           |
| 1/32  | 0.000437618605996000 | 0.000819599351780000           |
| 1/64  | 0.000010959389317000 | 0.000020415539641000           |
| 1/128 | 0.000002719908078700 | 0.000005123291081100           |
| 1/256 | 0.000000587876159303 | 0.000000128045530736           |

Table 16. Convergence rate

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_{\lambda}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.89676</td>
<td>1.98272</td>
</tr>
<tr>
<td>1/8</td>
<td>1.96739</td>
<td>1.99422</td>
</tr>
<tr>
<td>1/16</td>
<td>1.98976</td>
<td>1.99820</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99751</td>
<td>1.99947</td>
</tr>
<tr>
<td>1/64</td>
<td>2.01054</td>
<td>1.99988</td>
</tr>
<tr>
<td>1/128</td>
<td>2.20997</td>
<td>2.00042</td>
</tr>
</tbody>
</table>


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  | 0.0002974560387080000 | 0.0000059444075653000 |
| 1/8  | 0.0000821499048279000 | 0.0000153638374791000 |
| 1/16 | 0.0000212032218845000 | 0.0000038950058119800 |
| 1/32 | 0.0000053529135258400 | 0.0000009782954445650 |
| 1/64 | 0.0000134163030705000 | 0.000002496988333550 |
| 1/128| 0.0000033368060802280 | 0.000000612494781019 |
| 1/256| 0.0000007480663927975 | 0.00000015398376669 |

### Table 18. Convergence rate

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.85634</td>
<td>1.95157</td>
</tr>
<tr>
<td>1/8</td>
<td>1.95308</td>
<td>1.98027</td>
</tr>
<tr>
<td>1/16</td>
<td>1.98589</td>
<td>1.99328</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99634</td>
<td>1.99802</td>
</tr>
<tr>
<td>1/64</td>
<td>2.00742</td>
<td>1.99948</td>
</tr>
<tr>
<td>1/128</td>
<td>2.15725</td>
<td>2.00024</td>
</tr>
</tbody>
</table>

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 | 0.0000046308548170400 |
| 1/8  | 0.00171552692575000 | 0.0000149737518910000 |
| 1/16 | 0.0003798009532400 | 0.0000034316210158970 |
| 1/32 | 0.000005016553072700 | 0.0000004120205288900 |
| 1/64 | 0.0000012579927901520 | 0.000000002900370855020 |
| 1/128| 0.00000033593583475 | 0.0000000007286341007 |
| 1/256| 0.00000008597889469 | 0.000000000182513611470 |

### Table 20. Convergence rate

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.82426</td>
<td>1.79964</td>
</tr>
<tr>
<td>1/8</td>
<td>1.93205</td>
<td>1.86887</td>
</tr>
<tr>
<td>1/16</td>
<td>1.97641</td>
<td>1.94561</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99269</td>
<td>1.98250</td>
</tr>
<tr>
<td>1/64</td>
<td>2.10051</td>
<td>1.99511</td>
</tr>
<tr>
<td>1/128</td>
<td>2.12707</td>
<td>1.99891</td>
</tr>
</tbody>
</table>
3.11. Solve finite element equations as one equation by FEM. Matrix form of stationary point equations

\[ \begin{align*}
(\varphi, u)_{\Gamma_0} + 0 &= - (\nabla \varphi, \nabla \lambda)_{\Omega} - (\varphi, \lambda)_{\Omega} = (\varphi, u_0)_{\Gamma_0} \quad \text{for all } \varphi \in H^1(\Omega) \\
0 + \alpha (\chi, q)_{\Omega} &= (\chi, \lambda)_{\Omega} = 0 \quad \text{for all } \chi \in L^2(\Omega) \\
-(\nabla u, \nabla \psi)_{\Omega} - (u, \psi)_{\Omega} + (q, \psi)_{\Omega} &= 0 = 0 \quad \text{for all } \psi \in H^1(\Omega)
\end{align*} \]

The equations in matrix form

\[ \begin{bmatrix}
A_{\Gamma_0}^T u + 0 - (B_{\Omega}^T + A_{\Omega}^T) \lambda &= A_{\Gamma_0}^T u_0 \\
0 + \alpha A_{\Omega}^T q + A_{\Omega}^T \lambda &= 0 \\
-(B_{\Omega} + A_{\Omega}) \bar{u} + A_{\Omega} \bar{q} + 0 &= 0
\end{bmatrix} \begin{bmatrix}
u \\
q \\
\lambda
\end{bmatrix} = \begin{bmatrix}
A_{\Gamma_0}^T u_0 \\
0 \\
0
\end{bmatrix} \]

The block form of the matrix is:

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

where

\[ A_e = \begin{bmatrix}
A_{\Gamma_0}^T \\
0
\end{bmatrix}, B_e = \begin{bmatrix}
-(B_{\Omega} + A_{\Omega}) & A_{\Omega}
\end{bmatrix}, \]

\[ B_e^T = \begin{bmatrix}
-(B_{\Omega}^T + A_{\Omega}^T) \\
A_{\Omega}^T
\end{bmatrix}, b_e = \begin{bmatrix}
A_{\Gamma_0}^T u_0
\end{bmatrix} \]

Thus the block form of the matrix has eigenvalues \( \frac{A_e \pm \sqrt{A_e^2 + B_e B_e^T}}{2} \), one positive, one negative, therefore it is a saddle point.
Numerical form of the matrices, when $\alpha = 1$ with mesh $(2,1)$

$$
A_e = \begin{bmatrix}
A^T_{\Omega} & 0 \\
0 & \alpha A^T_{\Omega}
\end{bmatrix} =
\begin{bmatrix}
.5 & .083 & 0 & .167 & 0 & 0 & 0 & 0 & 0 & 0 \\
.083 & .33 & .083 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & .083 & .5 & 0 & .167 & 0 & 0 & 0 & 0 & 0 \\
.167 & 0 & 0 & .33 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & .167 & 0 & 0 & .33 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .083 & .0208 & 0 & .0208 \\
0 & 0 & 0 & 0 & 0 & 0 & .0208 & 0 & .0208 & .042 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0208 & .0208 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0208 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$A_e$ is positive definite.

$$
B_e = \begin{bmatrix}
-(B^T_{\Omega} + A^T_{\Omega}) & A^T_{\Omega}
\end{bmatrix} =
\begin{bmatrix}
-1.3 & .98 & 0 & .23 & -.042 & 0 \\
.98 & -2.6 & .98 & 0 & .45 & -.042 \\
0 & .98 & -1.3 & 0 & 0 & .23 \\
.23 & 0 & 0 & -1.3 & .98 & 0 \\
-.042 & .4583 & 0 & .98 & -2.6 & .98 \\
0 & -.042 & .23 & 0 & .98 & -1.3 \\
\end{bmatrix}
$$

$B^T_e = \begin{bmatrix}
-(B^T_{\Omega} + A^T_{\Omega}) \\
A^T_{\Omega}
\end{bmatrix} =
\begin{bmatrix}
-.133 & .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 & -.042 & .23 & 0 & .98 & -1.3 \\
\end{bmatrix} \
\begin{bmatrix}
.0833 & .02083 & 0 & .02083 & 0 & .042 & 0 \\
.02083 & .125 & .02083 & 0 & .042 & .042 \\
.02083 & .04167 & .02083 & 0 & .02083 & .02083 \\
.02083 & 0 & .04167 & .02083 & 0 & .02083 \\
.042 & .042 & 0 & .02083 & .125 & .02083 \\
0 & .04167 & .02083 & 0 & .02083 & .0833 \\
\end{bmatrix}
$$

$$
, b_e = \begin{bmatrix}
A^T_{\Omega} u_0^T
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 .25 \\
0 .5 \\
0 .25 \\
\end{bmatrix}
$$

$B^T_e$ is the transport of $B_e$.

Add these three equations into one equation:

$$
(\nabla \psi, \nabla u)_\Omega + (\nabla \varphi, \nabla \lambda)_\Omega + (\psi, u)_\Omega + (\varphi, \lambda)_\Omega + \alpha (\chi, q)_\Omega + (\chi, \lambda)_\Omega - (\psi, q)_\Omega - (\varphi, u)_\Gamma = 0, 
$$

for all $\psi, \varphi \in V$ and $\chi \in Q$.

Finite element equation:

$$
(\nabla \psi_h, \nabla u_h)_\Omega + (\nabla \varphi_h, \nabla \lambda_h)_\Omega + (\psi_h, u_h)_\Omega + (\varphi_h, \lambda_h)_\Omega + \alpha (\chi_h, q_h)_\Omega + (\chi_h, \lambda_h)_\Omega - (\psi_h, q_h)_\Omega - (\varphi_h, u_h)_\Gamma = 0, 
$$

for all $\psi_h, \varphi_h \in V_h$ and $\chi_h \in Q_h$.

Choose $u_0 = 1$, keeping $(u_h, q_h, \lambda)$ are variables.
Compute the objective function corresponding to various meshes and $\alpha$ values, it shows that value of objective function is decreasing as $\alpha$ is decreasing.

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

| $\alpha$ | $\frac{1}{2} ||u_h - u_0||^2$ | $\frac{2}{3} |q_h|^2$ | $J(u_h, q_h)$ |
| --- | --- | --- | --- |
| 1.0 | 0.50874855461900 | 0.1249892620210000 | 0.63373781664000 |
| 1/10 | 0.02261271860870 | 0.0460934834970000 | 0.06331216695850 |
| 1/100 | 0.00189580623560 | 0.0048132555633400 | 0.006706179894 |
| 1/1000 | 0.00111127114959 | 0.0004898622862820 | 0.0016026334588 |
| 1/10000 | 0.00104509048393 | 0.0000490747675328 | 0.00109416525146 |

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

| $\alpha$ | $\frac{1}{2} ||u_h - u_0||^2$ | $\frac{2}{3} |q_h|^2$ | $J(u_h, q_h)$ |
| --- | --- | --- | --- |
| 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 | 0.001118887670246 | 0.0004899125415890 | 0.00160178124405 |
| 1/10000 | 0.00104835647857 | 0.0000490782205093 | 0.00109743469907 |

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

| $\alpha$ | $\frac{1}{2} ||u_h - u_0||^2$ | $\frac{2}{3} |q_h|^2$ | $J(u_h, q_h)$ |
| --- | --- | --- | --- |
| 1.0 | 0.508682222633200 | 0.1249894328800000 | 0.63367165921300 |
| 1/10 | 0.02257100503590 | 0.0407041120386000 | 0.06327511707450 |
| 1/100 | 0.00188817453997 | 0.0048139064161000 | 0.00670208095607 |
| 1/1000 | 0.00111435647857 | 0.000489927001483 | 0.00160127851863 |
| 1/10000 | 0.00104913644090 | 0.000049079349339 | 0.00109821579024 |

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

| $\alpha$ | $\frac{1}{2} ||u_h - u_0||^2$ | $\frac{2}{3} |q_h|^2$ | $J(u_h, q_h)$ |
| --- | --- | --- | --- |
| 1.0 | 0.50867854031900 | 0.1249894421440000 | 0.633679824360 |
| 1/10 | 0.022568222633200 | 0.0407043783410000 | 0.06327300351190 |
| 1/100 | 0.00188768288631 | 0.0048139436942000 | 0.00670162685573 |
| 1/1000 | 0.00111435647857 | 0.000489927001483 | 0.00160127851863 |
| 1/10000 | 0.00104913644090 | 0.000049079349339 | 0.00109821579024 |
The objective function value is decreasing as \( \alpha \) is decreasing corresponding to various meshes.

Table 25. Objective function for various \( \alpha, h = \frac{1}{256} \)

| \( \alpha \) | \( \frac{1}{2} \| u_h - u_0 \|^2 \) | \( \frac{1}{2} | q_h |^2 \) | \( J(u_h, q_h) \) |
|---|---|---|---|
| 1.0 | 0.50867854031900 | 0.1249894421440000 | 0.63366798246300 |
| 1/10 | 0.02256862517100 | 0.0407043783410000 | 0.063273030351190 |
| 1/100 | 0.001887682868631 | 0.0048139439694200 | 0.00670162685573 |
| 1/1000 | 0.00011114834062 | 0.0004899308469340 | 0.00160107918756 |
| 1/10000 | 0.0000104912452421 | 0.0000490796990666 | 0.00109820422328 |

Table 26. Objective function for various \( \alpha, h = \frac{1}{128} \)

| \( \alpha \) | \( \frac{1}{2} \| u_h - u_0 \|^2 \) | \( \frac{1}{2} | q_h |^2 \) | \( J(u_h, q_h) \) |
|---|---|---|---|
| 1.0 | 0.50867854031900 | 0.1249894421440000 | 0.63366798246300 |
| 1/10 | 0.02256862517100 | 0.0407043783410000 | 0.063273030351190 |
| 1/100 | 0.001887682868631 | 0.0048139439694200 | 0.00670162685573 |
| 1/1000 | 0.00011114834062 | 0.0004899308469340 | 0.00160107918756 |
| 1/10000 | 0.0000104912452421 | 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 \( \alpha, h = \frac{1}{256} \)

| \( \alpha \) | \( \frac{1}{2} \| u_h - u_0 \|^2 \) | \( \frac{1}{2} | q_h |^2 \) | \( J(u, q) \) |
|---|---|---|---|
| 1.0 | 0.50867854031900 | 0.1249894421440000 | 0.63366798246300 |
| 1/10 | 0.02256862517100 | 0.0407043783410000 | 0.063273030351190 |
| 1/100 | 0.001887682868631 | 0.0048139439694200 | 0.00670162685573 |
| 1/1000 | 0.00011114834062 | 0.0004899308469340 | 0.00160107918756 |
| 1/10000 | 0.0000104912452421 | 0.0000490796990666 | 0.00109820422328 |

Table 28. Objective function for various \( \alpha \), exact solution

| \( \alpha \) | \( \frac{1}{2} | u - u_0 |^2 \) | \( \frac{1}{2} | q |^2 \) | \( J(u, q) \) |
|---|---|---|---|
| 1.0 | 0.50867115387900 | 0.1249894421440000 | 0.63366798246300 |
| 1/10 | 0.022555557251240 | 0.0411284104767000 | 0.06368398200210 |
| 1/100 | 0.0018739439694200 | 0.0048139439694200 | 0.00670162685573 |
| 1/1000 | 0.00011114834062 | 0.0004899308469340 | 0.00160107918756 |
| 1/10000 | 0.0000104912452421 | 0.0000490796990666 | 0.00109820422328 |
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 values 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   | 0.0002590139797890000 | 0.001536071228090000 | 0.001536071228090000 |
| 1/8   | 0.0000798812633927000 | 0.000397157961062000 | 0.000397157961062000 |
| 1/16  | 0.0000204015643823000 | 0.000103677900880000 | 0.000103677900880000 |
| 1/32  | 0.0000051303422770300 | 0.00002517717218009000 | 0.00002517717218009000 |
| 1/64  | 0.0000012809490828300 | 0.0000063008902309900 | 0.0000063008902309900 |
| 1/128 | 0.0000003237313962400 | 0.000001576074300580 | 0.000001576074300580 |
| 1/256 | 0.0000000876797923794 | 0.00000039553360406 | 0.00000039553360406 |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_q$</th>
<th>$r_\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.90428</td>
<td>1.99445</td>
<td>1.99446</td>
</tr>
<tr>
<td>1/8</td>
<td>1.96918</td>
<td>1.99445</td>
<td>1.99446</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99018</td>
<td>1.99445</td>
<td>1.99446</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99648</td>
<td>1.99445</td>
<td>1.99446</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99109</td>
<td>1.99445</td>
<td>1.99446</td>
</tr>
<tr>
<td>1/128</td>
<td>1.88448</td>
<td>1.99445</td>
<td>1.99446</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$u_{\text{max}}$</th>
<th>0.504594507766</th>
<th>$u_{\text{min}}$</th>
<th>0.487529450441</th>
<th>$u_{h_{\text{max}}}$</th>
<th>0.504595074803</th>
<th>$u_{h_{\text{min}}}$</th>
<th>0.487529082467</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{\text{max}}$</td>
<td>0.650484891946</td>
<td>$q_{\text{min}}$</td>
<td>0.421549514180</td>
<td>$q_{h_{\text{max}}}$</td>
<td>0.650487490162</td>
<td>$q_{h_{\text{min}}}$</td>
<td>0.421546641055</td>
</tr>
<tr>
<td>$\lambda_{\text{max}}$</td>
<td>$-0.421546641055$</td>
<td>$\lambda_{\text{min}}$</td>
<td>$-0.421546641055$</td>
<td>$\lambda_{h_{\text{max}}}$</td>
<td>$-0.650484891946$</td>
<td>$\lambda_{h_{\text{min}}}$</td>
<td>$-0.650487490162$</td>
</tr>
</tbody>
</table>
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 values 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 | 0.0002852948610840000 | 0.0002852948610910000 |
| 1/8   | 0.0001473424904610000 | 0.0000745085805927000 | 0.0000745085805996000 |
| 1/16  | 0.0000376509756712000 | 0.0000188959906327000 | 0.0000188959905961000 |
| 1/32  | 0.0000094787214655800 | 0.0000047450384664000 | 0.0000047450385590900 |
| 1/64  | 0.0000023755628988000 | 0.0000011878365308300 | 0.0000011878354128000 |
| 1/128 | 0.0000005973966691400 | 0.0000002971344445200 | 0.0000002971356007430 |
| 1/256 | 0.0000001614376568833 | 0.0000000745447509566 | 0.0000000745401756646 |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_q$</th>
<th>$r_\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.90207</td>
<td>1.99494</td>
<td>1.99492</td>
</tr>
<tr>
<td>1/8</td>
<td>1.96841</td>
<td>1.99494</td>
<td>1.99492</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99892</td>
<td>1.99494</td>
<td>1.99492</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99643</td>
<td>1.99494</td>
<td>1.99492</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99150</td>
<td>1.99494</td>
<td>1.99492</td>
</tr>
<tr>
<td>1/128</td>
<td>1.88771</td>
<td>1.99494</td>
<td>1.99492</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$u_{\text{max}}$</th>
<th>$u_{\text{min}}$</th>
<th>$u_{h_{\text{max}}}$</th>
<th>$u_{h_{\text{min}}}$</th>
<th>$q_{\text{max}}$</th>
<th>$q_{\text{min}}$</th>
<th>$q_{h_{\text{max}}}$</th>
<th>$q_{h_{\text{min}}}$</th>
<th>$\lambda_{\text{max}}$</th>
<th>$\lambda_{\text{min}}$</th>
<th>$\lambda_{h_{\text{max}}}$</th>
<th>$\lambda_{h_{\text{min}}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.910598420727700</td>
<td>0.879802575744700</td>
<td>0.910599416268700</td>
<td>0.879801900082700</td>
<td>1.173874281640400</td>
<td>0.760734244963700</td>
<td>1.173879408070400</td>
<td>0.760729065000000</td>
<td>-0.0700754244963700</td>
<td>-0.1173874281640400</td>
<td>-0.070072906500000</td>
<td>-0.1173879408070400</td>
</tr>
</tbody>
</table>
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 values 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   | 0.00061269172116100 | 0.00036482149866600 | 0.0000364821498664000 |
| 1/8   | 0.0001646279362000  | 0.00099791269859500 | 0.0000997912698694000 |
| 1/16  | 0.00004102761465300 | 0.00025811256038500 | 0.0000258112560800000 |
| 1/32  | 0.000010600388451500 | 0.000065259307409000 | 0.0000065259307666000 |
| 1/64  | 0.000002655663121960 | 0.000016368912797900 | 0.0000163689143963000 |
| 1/128 | 0.000000667867352741 | 0.000004096464029010 | 0.0000040964729021400 |
| 1/256 | 0.000000179782478712 | 0.000001026539066710 | 0.0000010265778599000 |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_q$</th>
<th>$r_{\lambda}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.89595</td>
<td>1.99650</td>
<td>1.99654</td>
</tr>
<tr>
<td>1/8</td>
<td>1.96722</td>
<td>1.99650</td>
<td>1.99654</td>
</tr>
<tr>
<td>1/16</td>
<td>1.98980</td>
<td>1.99650</td>
<td>1.99654</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99645</td>
<td>1.99650</td>
<td>1.99654</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99197</td>
<td>1.99650</td>
<td>1.99654</td>
</tr>
<tr>
<td>1/128</td>
<td>1.99331</td>
<td>1.99650</td>
<td>1.99654</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$u_{\text{max}}$</th>
<th>$u_{\text{min}}$</th>
<th>$q_{\text{max}}$</th>
<th>$q_{\text{min}}$</th>
<th>$\lambda_{\text{max}}$</th>
<th>$\lambda_{\text{min}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.990277560795</td>
<td>0.956787019236</td>
<td>0.827299976991</td>
<td>0.1276599573608</td>
<td>-0.0082729997699</td>
<td>-0.01276599573608</td>
</tr>
<tr>
<td>$u_{\text{hmax}}$</td>
<td>$u_{\text{hmin}}$</td>
<td>$q_{\text{hmax}}$</td>
<td>$q_{\text{hmin}}$</td>
<td>$\lambda_{\text{hmax}}$</td>
<td>$\lambda_{\text{hmin}}$</td>
</tr>
<tr>
<td>0.990277560795</td>
<td>0.956787019236</td>
<td>0.827299976991</td>
<td>0.1276599573608</td>
<td>-0.0082729997699</td>
<td>-0.01276599573608</td>
</tr>
<tr>
<td>0.956782851689</td>
<td>0.956782851689</td>
<td>0.827294423161</td>
<td>0.827294423161</td>
<td>-0.0127659952017</td>
<td>-0.0127659952017</td>
</tr>
</tbody>
</table>
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 values 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 | 0.004630854817050000 | 0.0000046308548170400000     |
| $1/8$   | 0.000175526925750000 | 0.001497351891800000 | 0.0000014973518910000000      |
| $1/16$  | 0.000043798009532400  | 0.000431621016104000  | 0.0000004316210158970000      |
| $1/32$  | 0.000011016535072700  | 0.000114020252188000  | 0.0000001140202588900000      |
| $1/64$  | 0.000002759927901520  | 0.000029003704720600  | 0.0000002900370550200000      |
| $1/128$ | 0.000000693597883475  | 0.000007286342997680  | 0.0000000728634341000000      |
| $1/256$ | 0.000000185977869469  | 0.000001825152350560  | 0.0000000182513611470000      |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_q$</th>
<th>$r_\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/4$</td>
<td>1.89337</td>
<td>1.99718</td>
<td>1.99719</td>
</tr>
<tr>
<td>$1/8$</td>
<td>1.96971</td>
<td>1.99718</td>
<td>1.99719</td>
</tr>
<tr>
<td>$1/16$</td>
<td>1.99119</td>
<td>1.99718</td>
<td>1.99719</td>
</tr>
<tr>
<td>$1/32$</td>
<td>1.99697</td>
<td>1.99718</td>
<td>1.99719</td>
</tr>
<tr>
<td>$1/64$</td>
<td>1.99924</td>
<td>1.99718</td>
<td>1.99719</td>
</tr>
<tr>
<td>$1/128$</td>
<td>1.99897</td>
<td>1.99718</td>
<td>1.99719</td>
</tr>
</tbody>
</table>

**Table 40. Maximum minimum, $\alpha = \frac{1}{1000}$**

<table>
<thead>
<tr>
<th>$u_{max}$</th>
<th>0.999019173658</th>
<th>$u_{min}$</th>
<th>0.96523299450</th>
<th>$u_{h_{max}}$</th>
<th>0.999020119312</th>
<th>$u_{h_{min}}$</th>
<th>0.965232280315</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{max}$</td>
<td>1.287859596650</td>
<td>$q_{min}$</td>
<td>0.834602915486</td>
<td>$q_{h_{max}}$</td>
<td>1.287888280810</td>
<td>$q_{h_{min}}$</td>
<td>0.834597375314</td>
</tr>
<tr>
<td>$\lambda_{max}$</td>
<td>0.0008346029155</td>
<td>$\lambda_{min}$</td>
<td>0.0012878595970</td>
<td>$\lambda_{h_{max}}$</td>
<td>0.0008345973753</td>
<td>$\lambda_{h_{min}}$</td>
<td>0.0012878882800</td>
</tr>
</tbody>
</table>
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 values 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  | 0.000665683809315000   | 0.007924392485330000   | 0.0000000792439248530000 |
| 1/8  | 0.000175617793290000   | 0.005136947066030000   | 0.0000000313694706598000   |
| 1/16 | 0.000044576129674000   | 0.001004919821230000   | 0.0000001004919821280000   |
| 1/32 | 0.000011573852200000   | 0.000284476252159000   | 0.0000000284476252155000   |
| 1/64 | 0.00000279086302120000 | 0.000074610464725600   | 0.0000000074610464251200   |
| 1/128| 0.00000070967414074800 | 0.000018935399572100   | 0.0000000189353994876000   |
| 1/256| 0.00000018771221600800 | 0.000004754091192940   | 0.0000000047541015383500   |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_u$</th>
<th>$r_q$</th>
<th>$r_{\lambda}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.89517</td>
<td>1.99384</td>
<td>1.99384</td>
</tr>
<tr>
<td>1/8</td>
<td>1.98547</td>
<td>1.99384</td>
<td>1.99384</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99827</td>
<td>1.99384</td>
<td>1.99384</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99922</td>
<td>1.99384</td>
<td>1.99384</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99329</td>
<td>1.99384</td>
<td>1.99384</td>
</tr>
<tr>
<td>1/128</td>
<td>1.90082</td>
<td>1.99384</td>
<td>1.99384</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$u_{max}$</th>
<th>$u_{min}$</th>
<th>$q_{max}$</th>
<th>$q_{min}$</th>
<th>$\lambda_{max}$</th>
<th>$\lambda_{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999901830707</td>
<td>0.966085802614</td>
<td>0.966085582089</td>
<td>0.999902654089</td>
<td>0.966085088075</td>
<td>0.966085088075</td>
</tr>
<tr>
<td>1.288997510140</td>
<td>0.835340300886</td>
<td>1.28913241250</td>
<td>0.83534773992</td>
<td>0.83534773992</td>
<td>0.83534773992</td>
</tr>
<tr>
<td>-0.00083553403069</td>
<td>-0.000128899745</td>
<td>-0.000835347744</td>
<td>-0.000128899745</td>
<td>-0.000835347744</td>
<td>-0.000128899745</td>
</tr>
</tbody>
</table>
The error analysis of objective function which is decreasing with an approximate factor $\frac{1}{4}$ as meshes are decreasing corresponding to various $\alpha$ values when $\alpha = 1.0$, $\frac{1}{10}$. The convergence rate approaches to 2.

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

| $h$   | $|J(u, q) - J(u_h, q_h)|$ |
|-------|--------------------------|
| 1/4   | 0.0000076156819594000    |
| 1/8   | 0.00000192465649741000   |
| 1/16  | 0.00000049067991299000   |
| 1/32  | 0.00000012317015266600   |
| 1/64  | 0.000000030815804585900  |
| 1/128 | 0.00000000770289696472   |
| 1/256 | 0.00000000191753701806   |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.87539</td>
</tr>
<tr>
<td>1/8</td>
<td>1.97175</td>
</tr>
<tr>
<td>1/16</td>
<td>1.99413</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99891</td>
</tr>
<tr>
<td>1/64</td>
<td>2.00020</td>
</tr>
<tr>
<td>1/128</td>
<td>2.00615</td>
</tr>
</tbody>
</table>

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

| $h$   | $|J(u, q) - J(u_h, q_h)|$ |
|-------|--------------------------|
| 1/4   | 0.00000397156637255000   |
| 1/8   | 0.00000109744635070000   |
| 1/16  | 0.00000028253648572500   |
| 1/32  | 0.00000007123896254710   |
| 1/64  | 0.00000001785296233760   |
| 1/128 | 0.00000000446597971382   |
| 1/256 | 0.00000000111552105331   |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.85556</td>
</tr>
<tr>
<td>1/8</td>
<td>1.95764</td>
</tr>
<tr>
<td>1/16</td>
<td>1.98770</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99650</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99911</td>
</tr>
<tr>
<td>1/128</td>
<td>2.00126</td>
</tr>
</tbody>
</table>
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 $\alpha$ values when $\alpha = \frac{1}{100}, \frac{1}{1000}$. The convergence rate is also computed.

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

| $h$  | $|J(u, q) - J(u_h, q_h)||$     |
|------|-------------------------------|
| 1/4  | 0.00000075562266102300000     |
| 1/8  | 0.00000227601607669000000     |
| 1/16 | 0.00000610235603169000000     |
| 1/32 | 0.00000156263010770000000     |
| 1/64 | 0.00000039358099853900000     |
| 1/128| 0.0000000098605837314800000   |
| 1/256| 0.0000000024653309801600000   |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.73116</td>
</tr>
<tr>
<td>1/8</td>
<td>1.89907</td>
</tr>
<tr>
<td>1/16</td>
<td>1.96539</td>
</tr>
<tr>
<td>1/32</td>
<td>1.99924</td>
</tr>
<tr>
<td>1/64</td>
<td>1.99989</td>
</tr>
<tr>
<td>1/128</td>
<td>1.99999</td>
</tr>
</tbody>
</table>

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

| $h$  | $|J(u, q) - J(u_h, q_h)||$     |
|------|-------------------------------|
| 1/4  | 0.0000016411334521000000     |
| 1/8  | 0.0000078057116503000000     |
| 1/16 | 0.0000029909478915000000     |
| 1/32 | 0.0000007974604638510000     |
| 1/64 | 0.0000002082069157990000     |
| 1/128| 0.00000005271855301902000   |
| 1/256| 0.00000001322202303810000   |

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

<table>
<thead>
<tr>
<th>$h$</th>
<th>$r_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1.04814</td>
</tr>
<tr>
<td>1/8</td>
<td>1.48124</td>
</tr>
<tr>
<td>1/16</td>
<td>1.80680</td>
</tr>
<tr>
<td>1/32</td>
<td>1.93740</td>
</tr>
<tr>
<td>1/64</td>
<td>1.98164</td>
</tr>
<tr>
<td>1/128</td>
<td>1.99537</td>
</tr>
</tbody>
</table>
3.12. **Python program.** Dolfin contains Python library, used to write finite element programs.

```python
from dolfin import *

define the mesh as unit square.

# 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 $\Gamma_o$ and $\Gamma \setminus \Gamma_o$.

``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*ds,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()
p
print 'Aa:',Aa.array()
p
print 'B:',B.array()
p
print 'Bt:',BT.array()
p
print'L:',b.array()
p
Define the finite element solution, split solution variables into \( (u_h, q_h, \lambda_h) \).

```

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

```
#finite element solution

uh=interpolate(u,U)
q=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 \( q \).

```
We=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 \( \lambda \).

```
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 \((u_h, q_h, \lambda_h)\).

```python
# 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 \(L^2\).

```python
# 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 \( \|\lambda-\lambda_h\|\)
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 \(J(u_h, q_h)\).

```python
# 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)
```

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

```python
# Compute \(J(u_h, q_h)\)
print('J(u_h, q.h) : ', ue_u0+q_e)
```

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

```python
# 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)
```

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

```python
# Compute \(J(u, q)\)
print('J(u, q) : ', ue_u0+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


Introduction and Examples (2009).

Lecture Notes (November 10, 2009).

The FEniCS Book (2011).