

INFLUENCE OF FUNCTIONALS ON THE CONDITIONING OF WELL  
DETERMINED LEAST SQUARES FINITE ELEMENT FORMULATION

by

ANIRUDH RAJAGOPAL

Presented to the Faculty of the Graduate School of  
The University of Texas at Arlington in Partial Fulfillment  
of the Requirements  
for the Degree of

Master of Science in Mechanical Engineering

THE UNIVERSITY OF TEXAS AT ARLINGTON

DECEMBER 2016

Copyright © by ANIRUDH RAJAGOPAL 2016  
All Rights Reserved

## ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Brian Dennis for not just his belief and motivation but also for his advice during the period of my master's course. His immense knowledge and experience in the field helped me throughout my research and documentation of my work. I wish to thank Dr. Bo Wang and Dr. Kent Lawrence for their interest in my research and for taking time to serve as member of my thesis committee.

A special thanks to Siddarth Chintamani and Aditya Raman for helping me through the course of my research and the documentation of my work. I would like to extend my appreciation to my other lab-mates Alok Rege, Ashkan Akbariyeh, James Grisham, Manoj Kandukuri, Sandeep Patil and Vivek Nair for their valuable input. A special mention to all my teachers back in India and here at University of Texas, Arlington.

Finally, I would like to thank my parents, Rajagopal Venkataraman and Jaya Rajagopal, and my brother, Arpith Rajagopal for their continuous support and motivation.

December 7, 2016

## ABSTRACT

### INFLUENCE OF FUNCTIONALS ON THE CONDITIONING OF WELL DETERMINED LEAST SQUARES FINITE ELEMENT FORMULATION

ANIRUDH RAJAGOPAL, M.S

The University of Texas at Arlington, 2016

Supervising Professor: Brian H. Dennis

The Least-Squares Finite Element Method (LSFEM) is a numerical method for solving partial differential equations (PDE) approximately by minimizing the L-2 norm of the PDE residuals. Unlike the more common Galerkin approach, this method does not employ integration by parts to reduce the continuity requirements of the basis functions. Instead, the PDE is cast as a first-order system of differential equations, which allows for the solution of primary and secondary variables with the same order of accuracy. In addition, this approach always leads to a direct minimization problem and therefore not subject to the restrictive inf-sup condition and does not result in an indefinite system of equations like the Galerkin method. Despite these advantages, it has been noted by other researchers that the choice of mesh and numerical integration scheme results in an implicit weighting of the functionals. This leads solutions that are very mesh sensitive and linear systems of equations that are ill-conditioned. This first part of this research is focused on creating a perfectly determined problem whose solution is independent of any implicit weighting. This is accomplished in 2-D by selectively subdividing quadrilateral mesh elements into

triangles and employing reduced integration. This results in a discrete system with exactly the same number of equations as unknowns. A sensitivity analysis is used on the whole domain and elements which are most sensitive to a weight factor are split until the desired number of equations is reached. A 2-D LSFEM solver was developed for hybrid quadrilateral/triangle meshes to demonstrate the method for elliptic and hyperbolic-elliptic equations. The conjugate gradient method was used as a solver for the resulting system of equations. Results show the solution to the resulting well-determined system is independent of any user defined weights applied to the functionals. An optimized set of weights was then obtained to minimize the number of conjugate gradient iterations required to solve the linear system of equations.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	iii
ABSTRACT . . . . .	iv
LIST OF ILLUSTRATIONS . . . . .	viii
LIST OF TABLES . . . . .	x
Chapter	Page
1. Introduction . . . . .	1
2. Literature Review . . . . .	3
2.1 Poisson Equation . . . . .	3
2.2 Convection-Diffusion Equation . . . . .	4
2.3 Conjugate Gradient . . . . .	5
2.4 Sensitivity Analysis . . . . .	5
3. Numerical Representation and Motivation . . . . .	6
3.1 Mathematical Representation . . . . .	6
3.2 Motivation . . . . .	12
4. Methodology . . . . .	14
4.1 Posing the problem . . . . .	14
4.2 Sensitivity Analysis . . . . .	16
5. Problem setup, Results and Discussion . . . . .	19
5.1 Poisson Equation . . . . .	19
5.1.1 Case 1: . . . . .	22
5.1.2 Case 2: . . . . .	26
5.2 Convection-Diffusion Equation . . . . .	30

5.2.1	Case 1: . . . . .	32
5.2.2	Case 2: . . . . .	36
6.	Conclusion and Future Work . . . . .	41
6.1	Conclusion . . . . .	41
6.2	Future Work . . . . .	41
Appendix		
A.	Validation . . . . .	43
A.0.1	For Primary Variable $T$ : . . . . .	44
A.0.2	For Flux Variable $q_x$ : . . . . .	45
A.0.3	For Flux Variable $q_y$ : . . . . .	46
B.	Optimizer . . . . .	47
	REFERENCES . . . . .	49
	Biography . . . . .	52

## LIST OF ILLUSTRATIONS

Figure	Page
3.1 An example plot of objective function $f(x)$ in two directional search as per Shewchuk, et al [4]. . . . .	11
4.1 Example of splitting 2' quadrature on a domain with quadrilateral elements . . . . .	14
4.2 An example of the sensitivity map for the whole domain . . . . .	17
5.1 A $5 \times 5$ uniform square grid with constant normal fluxes on the boundaries	22
5.2 The plot of LSFEM solution with constant normal fluxes on boundaries	23
5.3 Sensitivity map of individual functionals . . . . .	24
5.4 A $5 \times 5$ uniform square grid with constant $T$ on the boundaries . . . . .	26
5.5 The plot of LSFEM solution with constant $T$ on boundaries . . . . .	27
5.6 Sensitivity map of individual functionals . . . . .	28
5.7 Flow through parallel plates . . . . .	30
5.8 Flow through parallel plates with constant wall temperature . . . . .	32
5.9 The plot of LSFEM solution with constant wall temperature . . . . .	33
5.10 Sensitivity map of individual functionals . . . . .	34
5.11 Sensitivity map of individual functionals . . . . .	35
5.12 Flow through parallel plates with constant flux at the plate boundary .	36
5.13 The plot of LSFEM solution with constant wall flux . . . . .	37
5.14 Sensitivity map of individual functionals . . . . .	38
5.15 Sensitivity map of individual functionals . . . . .	39
A.1 Comparison of approximate to exact solution for variable $T$ . . . . .	44



A.2	Comparison of approximate to exact solution for variable $q_x$ . . . . .	45
A.3	Comparison of approximate to exact solution for variable $q_y$ . . . . .	46

## LIST OF TABLES

Table	Page
5.1 Initial equal weights . . . . .	25
5.2 Optimized weights . . . . .	25
5.3 Initial equal weights . . . . .	29
5.4 Optimized weights . . . . .	29
5.5 Initial equal weights . . . . .	36
5.6 Optimized weights . . . . .	36
5.7 Initial equal weights . . . . .	40
5.8 Optimized weights . . . . .	40

## CHAPTER 1

### Introduction

Scientists have been utilizing finite element techniques to solve fluid and heat transfer problems for many decades now. Perhaps, the lack of a comprehensive finite element code, covering a large spectrum of problems is motivation to many enthusiasts.

The motivation behind this study is also to advocate such a common and more diverse technique namely, the least-squares finite element method. This study also explains certain complications behind using the least-squares technique in different genre of problems. For certain problems, the least-squares finite element method has conveniently yielded discrete linear systems that are symmetric and positive definite such that other methods have failed to do so.

The least-squares finite element method (LSFEM) that has been implemented here is essentially minimizing the  $L_2$  norm of the residuals of given system of partial differential equations. As this study uses  $C^0$  elements to solve the problem cases, second order 2D partial differential equations have to be minimized to a set of first order system formulations by introducing two flux variables. This study has been implemented on problems with simple boundary conditions and smooth solutions. Unfortunately, most applications in reality is not as simple as this, in that, often complex combinations of boundary conditions have to be used and the solutions might not be smooth enough to obtain good accuracy. The focus of this work is to implement the method efficiently.

There are many advantages of the LSFEM namely,

- Avoiding the LadyzhenskayaBabuskaBrezzi (LBB) condition by using a common approximate function.
- The choice of this function will also not affect the LBB condition.
- Implementing boundary conditions is simpler.
- Essentially reducing the order of system of equations, thereby reducing the complexity and using simpler approximate functions.

One of the few problems faced by this technique is that the condition number of the system of equations that would be generated depends extensively on the type of problem. Thus, the technique was considered fruitful for only certain types of problems. Another problem with LSFEM is that the technique very rarely solves a well-determined system of equations. The number of unknowns to the number of equations have to match in order to best pose the problem.

## CHAPTER 2

### Literature Review

In order to solve a governing partial differential equation in the least-squares sense, its residual along with that of the boundary conditions has to be approximated. With a long history shown by Eason [5], it is still a widely used finite element method. Boundary valued problems have been widely solved in the past as shown in work by Bramble and Schatz [6] and this has been used by Varga [7].

The least-squares technique can be compared to traditional Galerkin Finite Element techniques as described by Jiang and Povinelli [3]. Lynn and Arya [8] and Zienkiewicz, et al [9] showed how introducing an additional unknown variable can reduce the order of the governing PDE and also maintain smoothness using  $C^0$  elements. Straight forward problems were first solved in Lynn [10], Lynn and Alani [11], Polk and Lynn [12] and Fix and Guzburger [13]. This progressed onto a generalized Cauchy-Reimann equation, where an error estimate is provided by Fix and Rose [14]

#### 2.1 Poisson Equation

Petrovsky type elliptic equations were solved theoretically by Wendland [15]. Petrovsky systems are a class of elliptic systems where the all the functionals and the unknown variables are of the same continuity. Another general method was introduced by Aziz, et al [16]. The method introduces the Agmon-Douglis-Nirenberg (ADN) type elliptic equations where the weighted residuals of the functionals is reduced.

The work done by Jiang, et al [3] has been an appropriate start for this thesis. In the

paper [3], the author compares the various finite element technique and formulations for simple elliptic problem. The study introduces a new optimal least-squares technique where the compatibility condition (irrotationality) was introduced and this , as claimed, proves that the ellipticity of the system to be regained. The paper also solves for a forcing function derived through the method of manufacturing. This method is a way of arriving at a forcing function keeping the analytical solution as a fixed entity. This thesis also utilizes the same forcing function throughout the Poisson equation study. Even the least-squares solutions were validated with respect to this forcing function.

## 2.2 Convection-Diffusion Equation

The work done by Fletcher [17] is a good reference to solve for energy-free Euler equation. This thesis has a lot of commonality with the work done by Nguyen and Reynen [18], in that, both solve for the advection-diffusion equation using least-squares technique but on a rather different bandwidths of Peclet number. Also, this study is purely based on proving that the new technique developed works right on a wide variety of problems. The upwinding scheme used in [18] is through the Taylor-Galerkin model, which is essentially rendered useless because of the least-squares formulation extended into the time domain.

The main idea about heat transfer between parallel plates has been derived from Sellers, et al [19] and Haji-Sheikh, et al [20]. The boundary conditions and exact solution has been computed from the same. These two papers represent the pure theoretical idea of the heat transfer problem.

### 2.3 Conjugate Gradient

This study focuses on estimating the contributions of individual functionals to the overall condition of the system. In order to preserve this contributions, no pre-conditioner of any kind was used with the iterative solver - conjugate gradient technique. The conjugate-gradient algorithm and methodology was derived from the book by Saad [4]. This book proved very vital for the understanding of vector notations, vector calculus and the algorithms of various iterative solvers. The works by Jiang [1-3],[21-26] and Carey [26] all utilize the same conjugate-gradient algorithm, either with or without the pre-conditioner. The common pre-conditioner used would be the LU pre-conditioner.

### 2.4 Sensitivity Analysis

Majority of the finite element techniques are chosen according to problem because of the condition of the system of equations. This is avoided by estimating the condition of the system through the sensitivity analysis. Grcar [27] provides a realistic way of relating the sensitivity to the condition number of a linear system. Luo and Tseng [28] have shown what the bound on the distance from any point to the solution set of a linear system.

## CHAPTER 3

### Numerical Representation and Motivation

This section consists of two parts - the numerical formulation and the motivation inspired from these formulations. The least-squares formulations is an effective way of generating symmetric positive definite stiffness matrices for many non self-adjoint problems, in which the governing equations are reduced to a first order system of equations. The systems that permit the least-square functional representation in  $H^1$  - norm equivalent are termed  $H^1$  - coercive formulations.

#### 3.1 Mathematical Representation

Most governing equations in fluid dynamics and heat transfer problems generally are second order partial differential equations. The 2D form of these equations can be represented in terms of a first order *non-dimensionalized* system of equations by introducing two flux terms in each direction. The solution for all the variables can be determined simultaneously. For simplicity, this study involves only steady state problems.

Consider a boundary valued scalar transport problem,

$$L\hat{u} = \hat{f} \quad \text{in } \Omega \quad (3.1)$$

$$B\hat{u} = \hat{g} \quad \text{in } \Gamma \quad (3.2)$$

Here  $L$  is a first order partial operator and varies with problem such that,

$$L\hat{u} = \sum_{i=1}^n \alpha \frac{\partial \hat{u}}{\partial x_i} + \beta \hat{u} \quad (3.3)$$



The domain is bounded such that  $\Omega \in \mathbb{R}^n$  with a piecewise smooth boundary  $\Gamma$  where  $n$  represents the number of linearly independent vectors or simply just the number of dimensions in this case.  $\hat{u}^T = (u^1, u^2, u^3, \dots, u^m)$  is a vector of unknown functions.  $\alpha$  and  $\beta$  are merely square matrices which determine the type of problem and  $\hat{f}$  represents the force vector.  $B$  is a boundary operator and  $\hat{g}$  represents the force values at the boundaries.

The main point is to notice is this is a generalized way of formulating the least-squares system. The operators mentioned in (3.1) will indirectly reflect the conditionality of the system of equations as they depend on the problem definition. This problem maybe caused by many factors say, if the problem is a non-self adjoint Sturm-Louville type or if the operators are not scaled right. This study assumes that (3.1) has a unique smooth (asymptotic) solution and the approximate function is chosen such that it captures the solution well [1,2].

The common notation  $\hat{L}^2(\omega)$  represents the L-2 norm of the vector, in that, it denotes the space for square-integrable functions within the domain  $\omega$  such that the inner product is given by,

$$(u, v) = \int_{\Omega} uv \, d\omega \quad u, v \in \hat{L}^2(\Omega) \quad (3.4)$$

and the norm,

$$\|u\|_0^2 = (u, u) \quad u \in \hat{L}^2(\Omega) \quad (3.5)$$

For 'm' number of unknowns in  $\hat{u}$ , there are two spaces,

$$L^2(\Omega) = (\hat{L}^2(\Omega))^m \quad (3.6)$$

$$H^1(\Omega) = (\hat{H}^1(\Omega))^m \quad (3.7)$$

Where  $\hat{H}^1(\Omega)$  and  $H^1(\Omega)$  represent the Sobolev sub-spaces for each component and for all the components respectively.

Since the least squares minimizes the residual function, least squares functional set has to be generated as follows,

$$R(\hat{u}) = \|L\hat{u} - \hat{f}\|_0^2 = (L\hat{u} - \hat{f}, L\hat{u} - \hat{f}) \quad (3.8)$$

The left hand side represents the residual function or set of the individual functionals and it varies according to the reduced first order system of equations. The weak statement can be written in terms of the approximate function such that  $\hat{u} \in \vec{S}$ ,

$$(L\hat{w}, L\hat{u}) = (L\hat{w}, L\hat{f}) \quad (3.9)$$

Where function space  $S = \{\hat{u} \in (\hat{H}^1(\Omega))^m; B\hat{u} = 0 \text{ on } \Gamma\}$ ,  $\hat{w} = \delta\hat{u}$

The weak statement can be written in the form of the approximate function. This is done by first discretizing the domain according in finite number of elements and then mapping the solutions onto the Gauss points using the shape function as shown below,

$$\hat{u}(x) = \sum_{i=1}^{N_e} \psi_i(x) \hat{u}_i \quad (3.10)$$

Where  $\hat{u}_i$  represents node-wise unknowns. Upon substituting this approximation onto the weak statement, a linear system of equation is formed in terms of stiffness

matrix  $\mathbf{K}$  which is assembled from elemental stiffness matrices  $K_e$ , unknown vector  $\mathbf{U}$  and load vector  $\mathbf{F}$  assembled from elemental force vectors  $F_e$

$$\boxed{\mathbf{KU} = \mathbf{F}} \quad (3.11)$$

The elemental stiffness matrices and load vectors are represented in terms of respective elemental shape function  $\psi = (\psi_1, \psi_2, \psi_3, \dots, \psi_{N_e})^T$  as,

$$\mathbf{K}_e = \int_{\Omega_e} \{L\psi\}^T \{L\psi\} d\Omega \quad (3.12)$$

$$\mathbf{F}_e = \int_{\Omega_e} \{L\psi\}^T \hat{f} d\Omega \quad (3.13)$$

The stiffness matrix  $\mathbf{K}$  is symmetric positive definite in LSFEM, as such, iterative methods like Conjugate Gradient technique can be used to invert  $\mathbf{K}$  and estimate the unknown vector  $\mathbf{U}$  accurately. As each of the reduced first order system of equations produces a functional term in least-squares form, the weights of these individual functionals determine their contribution to the condition number of the stiffness matrix. Since inverting a stiffness matrix through iterative techniques like the conjugate gradient will depend on its condition number, it is very vital that choosing the weights of these individual functionals will determine the rate of convergence of the iterative solver (the number of iterations).

Inverting the stiffness matrix using conjugate gradient (CG) method is one of the most widely used iterative scheme. In order to preserve the effect of the LSFEM on the condition number of the stiffness matrix, no pre-conditioners were used in this study. CG is used essentially to solve a linear system of equations by starting with an initial guess and progressing iteratively towards to solution. In a generalized way,

the CG method can be written in its quadratic form by representing it in a scalar quadratic form of a vector.

$$f(x) = \frac{1}{2}\vec{x}^T A\vec{x} - \vec{b}^T \vec{x} + c \quad (3.14)$$

where  $f(x)$  is the objective function,  $A$  is the matrix operator that has to be inverted,  $\vec{x}$  is the unknown vector,  $\vec{b}$  is a vector and  $c$  is a scalar constant. In order for CG method to be used, the matrix operator  $A$  has to be symmetric and positive definite.

For a matrix operator  $A$  to be symmetric and positive definite,  $\vec{x}^T A\vec{x} > 0$ . The gradient of a vector field at a given point, gives the magnitude of its change in particular direction.

$$f'(x) = \left[ \frac{\partial}{\partial x_1} f(x) \frac{\partial}{\partial x_2} f(x) \frac{\partial}{\partial x_3} f(x) \frac{\partial}{\partial x_n} f(x) \right]^T \quad (3.15)$$

The goal of this method is to minimize the objective function  $f'(x)$ . Since  $A$  is symmetric, (3.14) can be reduced to the form

$$f'(x) = A\vec{x} - \vec{b} \quad (3.16)$$

Upon setting the gradient to zero, (3.15) can be written in terms of a linear system of equations

$$A\vec{x} - \vec{b} = 0 \quad (3.17)$$

In conjugate gradient we follow the path of steepest descent, where it is initialized at an arbitrary point  $x_0$  and iteratively reaches the minimum. This process cycles over till a tolerance limit is reached either in maximum iteration limit set by the user or a minimum residual between two iterations. At the beginning of each iteration, the direction will first be calculated such that it provides the shortest path to the solution.

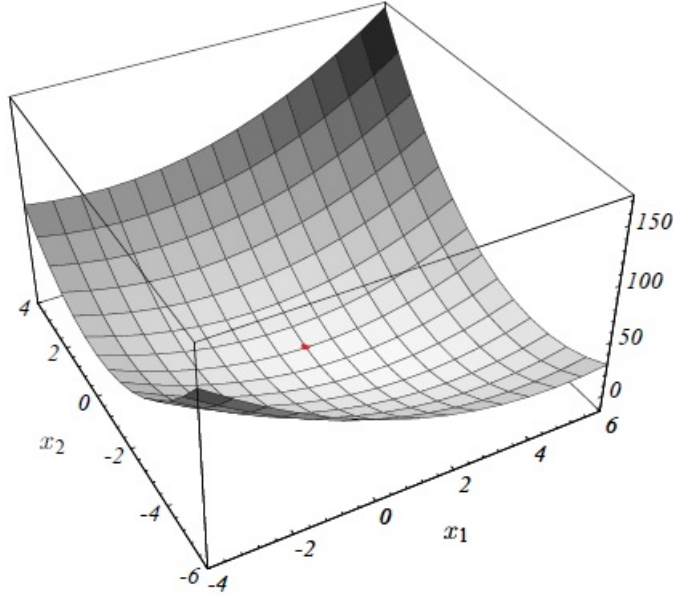


Figure 3.1: An example plot of objective function  $f(x)$  in two directional search as per Shewchuk, et al [4].

The above plot represents an example of the optimization search space for minimizing the objective function  $f(x)$  and the red dot represents its minimum. Line search helps choose the  $\alpha$  to minimize  $f(x)$  along a line but how big should its magnitude be in order to be represented in the form,

$$x_1 = x_0 + \alpha r_0 \tag{3.18}$$

The residual  $r_i$  is calculated at the end of each iteration to determine the distance to solution. The residual in reality provides a sense of direction for the next step.  $\alpha$  denotes the direction that minimizes  $f(x)$  if  $\frac{df(x_{(1)})}{d\alpha} = 0$ . From this, the value of  $\alpha$  can be calculated. The slope of the parabola in Figure 3.1 at any point represents the projection of the gradient onto the line such that  $f(x)$  is minimized at a point where the gradient is orthogonal to the search line. From this it is clear that we have to start with assuming that the residual of each iteration is orthogonal to the previous.

$$r_1^T r_0 = 0 \tag{3.19}$$

$$(b - Ax_1)^T r_0 = 0 \tag{3.20}$$

$$(b - Ax_0 + \alpha r_0)^T r_0 = 0 \tag{3.21}$$

$$\alpha = \frac{r_0^T r_0}{r_0^T A r_0} \tag{3.22}$$

In order to reduce the matrix-vector product twice within each steepest decent step, (3.18) can be pre-multiplied with  $A$  as,

$$r_{i+1} = r_i - \alpha_i A r_i \tag{3.23}$$

In the LSFEM formulation the matrix operator  $A$  is  $K$ . Thus, in order to invert the stiffness matrix efficiently without stability issues, the condition number of the matrix has to be as close to 1 as possible.

### 3.2 Motivation

The analogy that has been put forth already is that the condition number of the stiffness matrix is very important while inverting the same using CG method. The condition number primarily depends on the type of problem - characteristics of the governing equation, type of boundary conditions and technique used to derive the formulations. But the weights on individual functionals can also affect the contribution of these functionals on the overall condition number of the system. This study primarily focuses on estimating this very contribution. An optimization study

on various fluid and heat transfer problems was conducted keeping in mind, the condition of the system as the objective to be minimized.

One of the main problems with least-squares scheme is that the problem becomes either a over-determined system or an under-determined system because of the degree of freedom associated with the problem. In order to pose the problem right, the number of equations to the number of unknowns for the system of equations has to be matched. This way, a more conditioned system is formed.

The first task from this point would be matching the number of equations to the number of unknowns. This is done by increasing or decreasing the number of Gauss points where the problem is solved before being numerically integrated using Gaussian Quadrature scheme for the entire domain. After this, an optimizer is used as a black box to minimize the condition number of the matrix to as close to one as possible.

## CHAPTER 4

### Methodology

The primary goal of this study was to tackle the conditionality problem associated with varying type of problems. This section talks in detail about how this is done. There are three main steps that were taken and they are enlisted in the upcoming subsections.

#### 4.1 Posing the problem

In most finite element formulations, the number of equations does not match with the number of unknowns. Here an example has been described in order to make this concept a little more clearer.

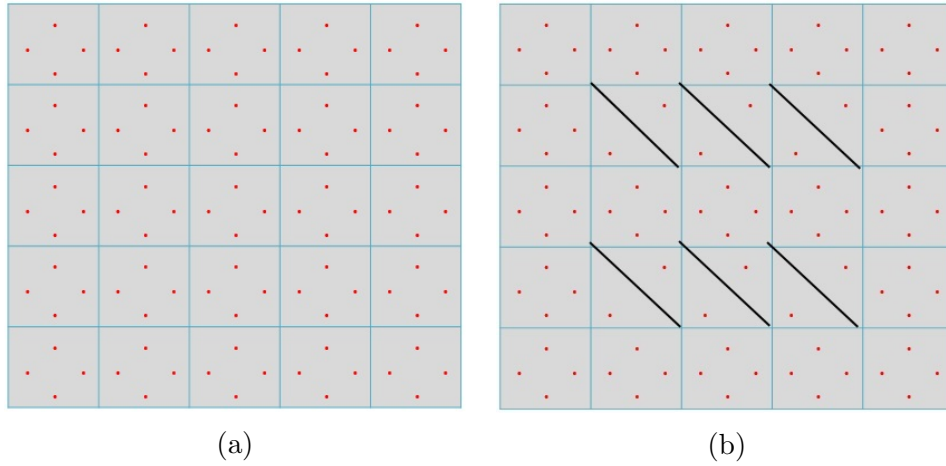


Figure 4.1: Example of splitting 2' quadrature on a domain with quadrilateral elements



Consider the number of elements within the domain as shown in Figure 4.1(a) to be a 5X5 uniform quadrilateral grid structure that uses 2 point Gaussian quadrature to solve for quadrilateral elements. Thus there are 4 Gauss points for each element where the solutions are numerically integrated for the whole domain, as shown in Figure 4.1. In a way, the problem is being solved onto these Gauss points and mapped onto the nodes.

For a simple potential flow, the governing equation can be represented in terms of div-curl-grad system of equations. This gives one functional for divergence and curl equations and one in each direction for the gradient term. Thus for a 2D problem as shown in Figure 4.1,

The number of functionals (A) = 4

The total number of Gauss points for the domain (B) = 100

Thus, total number of equations (M) = A.B = 400

The number of nodes (C) = 36

Variables for a div-curl-grad representative of potential flow would be scalar potential and flux in two directions.

The number of variables (D) = 3

Total number of unknowns (N) = C.D = 108

Thus there is an imbalance in the number of equations to the number of unknowns. Since, in this example there are more number of equations than unknowns, the system becomes over-determined. Similarly, if 1' quadrature is used to solve for the quadrilateral elements, system will become under-determined. Thus in order to pose the problem in a correct fashion, the total number of Gauss points have to be altered. A quadrilateral-split technique has been introduced in this study where certain quadrilateral elements are split in two triangular elements in which just 1' quadrature

is used as shown in Figure 4.1(b).

The number of elements and location of the elements that have to be split are the big questions. How many elements does it take to make system a well-determined one? Holding onto the notations from the above stated example,

$$\text{Total number elements to be split} = \frac{1}{2}\{M - N + \beta\} \quad (4.1)$$

Where  $\beta$  refers to the number of nodes on which Dirichlet boundary conditions have been applied. The  $\frac{1}{2}$  indicates that a single quadrilateral element will be split into 2 triangular element. The next question would be what are the locations at which the split has to be done? For this a sensitivity analysis is performed which is described in the next section.

## 4.2 Sensitivity Analysis

The location for splitting the quadrilateral elements is very important and need not be random in most cases. The rate of convergence has been monitored by varying the randomness of this location and was found that the location also affects the rate of convergence. A firm assumption that the rate of convergence or number of iterations needed for convergence is considered as a good indication of the condition number of the stiffness matrix that the CG inverts. This means that by finding the locations which sensitive, in terms of number of iterations, to perturbations to the weight factors of individual functionals can provide a good map of the location for split.

For this, a sensitivity analysis has been performed on each functional on a element by element basis. The weight factor of a functional has to be perturbed by such a value that neither the numerical error should not outweigh it nor should it be too big to make drastic change in the condition of the system.

Consider a system such that the condition number is dependent on the weights of the individual functionals,  $c = c(w_0)$  and  $c(w_0)$  number of iterations to convergence. Here  $w_0$  denotes the initial weight factor without being perturbed. After perturbation, according to Taylor series expansion:

$$c(w_0 + \Delta w) = c(w_0) + \Delta w \frac{dc}{dw} \Big|_{w_0} + \dots \quad (4.2)$$

Where  $\Delta w$  represents the sensitivity factor,  $w_0$  is the initial weight factor. Since all the functionals were initially given the same weight,  $c(w_0)$  remains constant for all cases in a problem.

$$Sensitivity = \frac{dc}{dw} \Big|_{w_0} = \frac{c(w_0 + \Delta w) - c(w_0)}{\Delta w} \quad (4.3)$$

The sensitivity value is calculated for each weight factor on each element. This gives a map of sensitivity for the whole domain per weight.

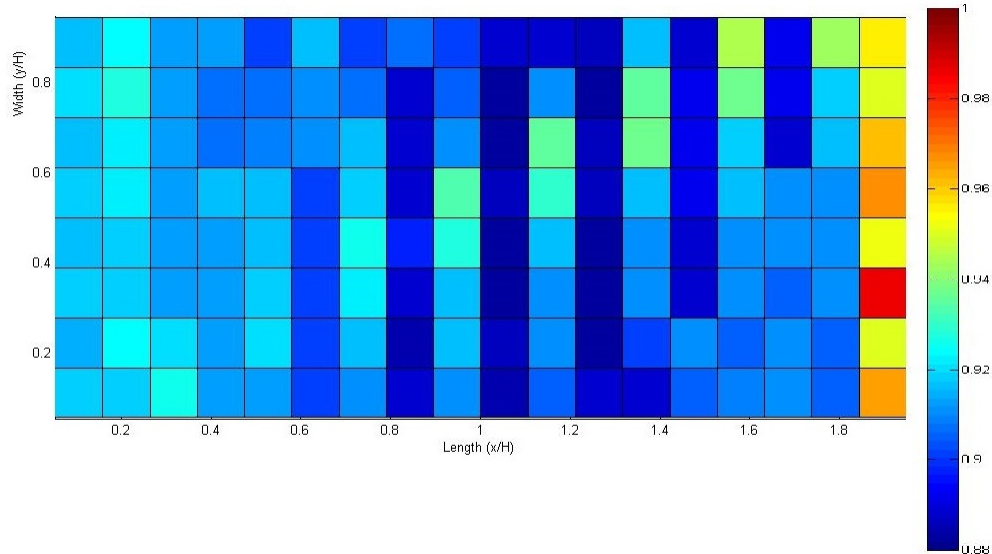


Figure 4.2: An example of the sensitivity map for the whole domain

The Figure 4.2 illustrates an example of how the sensitivity map would look like. This map indicates how good the locations that are most sensitive to weight factor changes. The maps are all normalized by the highest value in this study to get a cumulative estimate of the sensitivity. The maps for all functionals are put together and the most sensitive elements are picked. The number of elements picked is given in equation (4.1) and these are split. Another important information to be noted is that, Dirichlet boundary elements should not be split. This is because, even though these elements do contribute to the condition of the system, it is unavoidable. The Dirichlet boundary conditions are not calculated through the least-squares formulations but rather are hard coded into the stiffness matrix and load vector.

## CHAPTER 5

### Problem setup, Results and Discussion

The study involves investigation of the contribution of each functional on the overall condition of the system. It is imperative that the characteristics of the governing differential operator determines the condition of the system. Thus, in order to develop a more universal finite element code, it is very crucial to handle different kinds of ill-conditioned systems without using a pre-conditioner for the iterative solver but rather through the formulations itself.

With that as a motivation, the methodology that was discussed in the last chapter, has been implemented on two types of problems. Even within each of these problems, multiple situations have been implemented to observe a trend within a problem. This way, any future research can pick up where this study leaves.

#### 5.1 Poisson Equation

Diffusion of a scalar quantity is governed by the Poisson equation which essentially solves the PDE with a source term in terms of spacial dimensions. This study considers a 2-D steady incompressible Poisson equation in terms of any scalar variable  $T$ ,

$$\Delta^2 T = \hat{f} \tag{5.1}$$

The main advantage of using the least-squares sense is to reduce the order of the differential operator to  $L$  as shown in equation (3.1). The differential operator here in the Poisson equation is a second order PDE and it can be reduced to a first

order differential operator of the form equation (3.1) by introducing two variables in the form of fluxes ( $\vec{q}$ ). Once the flux is introduced, the second order PDE is reduced to a system of first order equations known as "*The Div-Curl-Grad System*" which are collectively solved,

$$\vec{q} = -k \Delta T \quad \text{in } \Omega \quad (5.2)$$

$$\nabla \cdot \vec{q} = \vec{f} \quad \text{in } \Omega \quad (5.3)$$

$$\nabla \times \vec{q} = 0 \quad \text{in } \Omega \quad (5.4)$$

Where  $k$  represents a material constant,  $\vec{q}$  represents the flux vector and  $\vec{f}$  represents the source term in the domain  $\Omega$ .

The above system of equations represent the Div-Curl-Grad system. Here equation (5.2) represents the gradient term where a flux term has been introduced in terms of the scalar variable. Since in this study, the domain is in 2-D, this term can be opened up into two terms. The equation (5.3) represents the Poisson equation in first order terms after introducing the fluxes or simply the divergence term. The equation (5.4) represents the curl of the fluxes. This essentially puts forth an irrotationality condition. This is to introduce an optimality in the least-squares formulation as suggested by Jiang and Povinelli [5].

There are three variables in this formulation, namely  $T$ ,  $\vec{q}_x$ ,  $\vec{q}_y$  and they can be mapped onto the transformation plane by introducing the shape function  $\psi$

$$T = \sum \psi_i T_i \quad (5.5)$$

$$q_x = \sum \psi_i q_{x_i} \quad (5.6)$$

$$q_y = \sum \psi_i q_{y_i} \quad (5.7)$$

It is clear from equations (5.2) - (5.4) that there are 4 functionals that have to be collectively minimized in the 2-D least-squares non-dimensionalized sense.

$$\pi = \pi_1 + \pi_2 + \pi_3 + \pi_4 \quad (5.8)$$

Where,

$$\pi_1 = \int_{\Omega} \left[ \vec{q}_x + \frac{\partial T}{\partial x} \right]^2 d\Omega \quad (5.9)$$

$$\pi_2 = \int_{\Omega} \left[ \vec{q}_y + \frac{\partial T}{\partial y} \right]^2 d\Omega \quad (5.10)$$

$$\pi_3 = \int_{\Omega} \left[ \frac{\partial \vec{q}_x}{\partial x} + \frac{\partial \vec{q}_y}{\partial y} - \vec{f} \right]^2 d\Omega \quad (5.11)$$

$$\pi_4 = \int_{\Omega} \left[ \frac{\partial \vec{q}_x}{\partial y} - \frac{\partial \vec{q}_y}{\partial x} \right]^2 d\Omega \quad (5.12)$$

This can be reduced in the form of a common first order operator as shown in equation (3.1) and represented as

$$\frac{\partial \{\pi\}}{\partial \{\hat{V}\}} = \sum_j \frac{\partial \{\pi_j\}}{\partial \{\hat{V}\}} = [L]\{u\} - \{f\} = [w]\{R\}^T \{u\} \{R\} - [w]\{R\}f \quad (5.13)$$

Where,  $\{\hat{V}\}$  represents the respective dimensions,  $\{\hat{R}\}$  represents a vector of functionals and  $[w]$  represents the elemental set of weight. This equation (5.13) has to be minimized to 0. The above equation (5.13) can be then reduced to form,

$$[K]\{U\} = \{F\} \quad (5.14)$$

### 5.1.1 Case 1:

Consider an example of a  $10 \times 10$  uniform square grid as shown in the Figure (5.1) below. The problem is solved for a source term given by Jiang and Povinelli [3] and the exact solutions have been shown in the validation chapter in comparison with the finite element solution. Here, only the problem definition and the simulation solution has been shown.

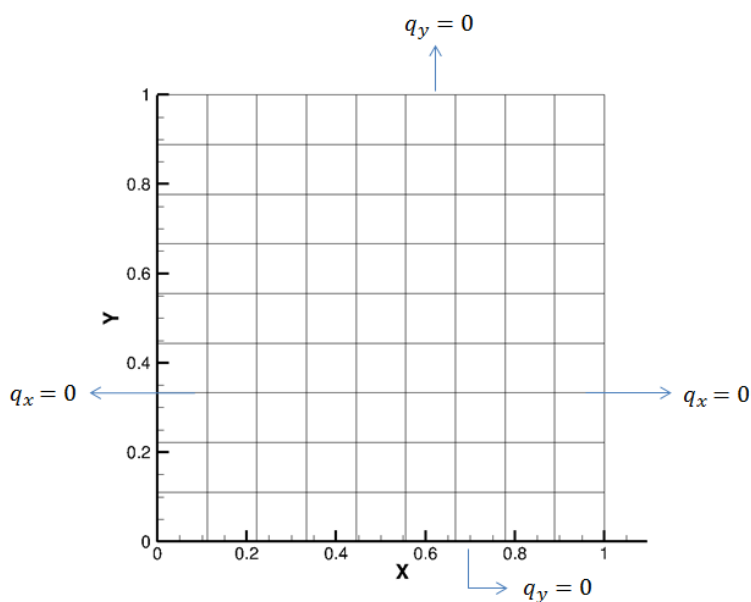


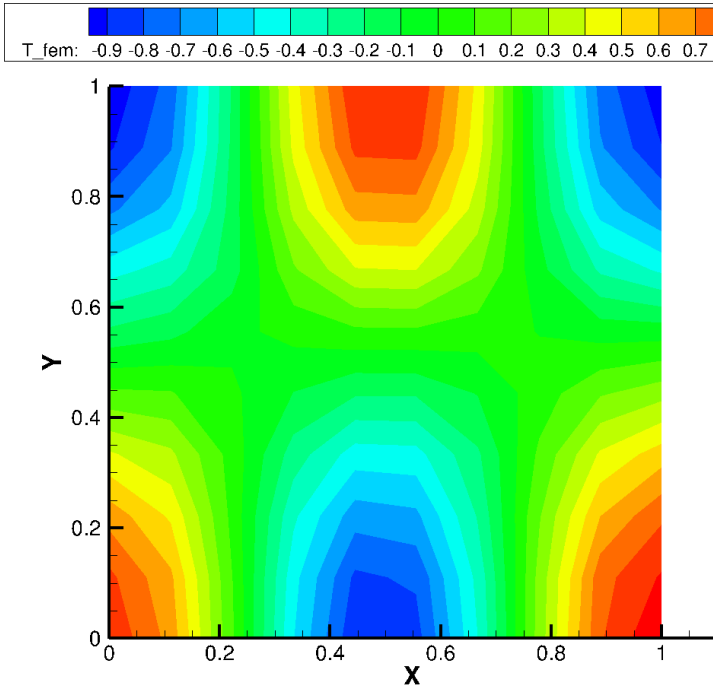
Figure 5.1: A  $5 \times 5$  uniform square grid with constant normal fluxes on the boundaries

According to [3], the source term was,

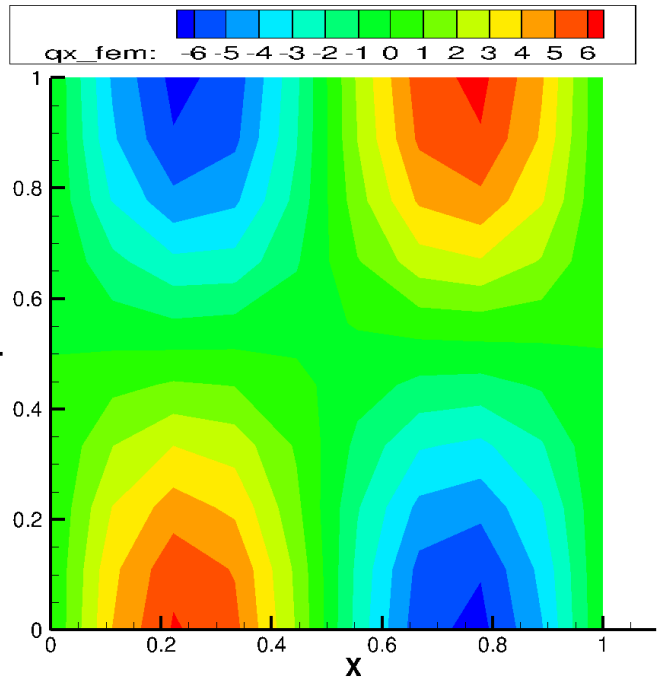
$$F = (5\pi^2 + 1)\cos(2\pi x)\cos(\pi y) \quad (5.15)$$

The plot below indicates the least-squares solution of  $T$ ,  $\vec{q}_x$  and  $\vec{q}_y$ .

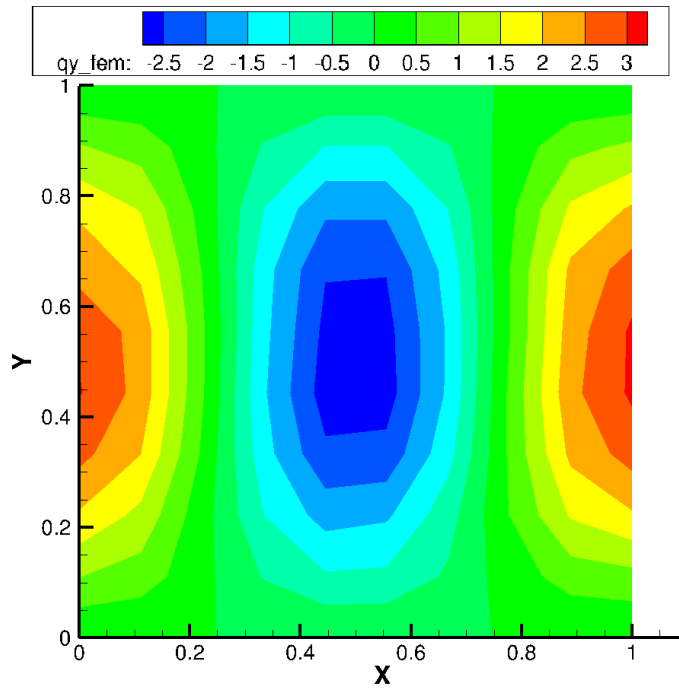




(a) LSFEM solution for  $T$



(b) LSFEM solution for  $q_x$



(c) LSFEM solution for  $q_y$

Figure 5.2: The plot of LSFEM solution with constant normal fluxes on boundaries

### 5.1.1.1 Sensitivity Analysis

Like discussed before, a sensitivity analysis was conducted on an element by element basis. The location to split the quadrilateral elements can be estimated from the sensitivity map. The weight factors were initially normalized to 1 ( $weights = [0.25 \ 0.25 \ 0.25 \ 0.25]$ ). Each weight was perturbed by  $10^{-6}$  and the number of iterations for each case was mapped and has been shown below.

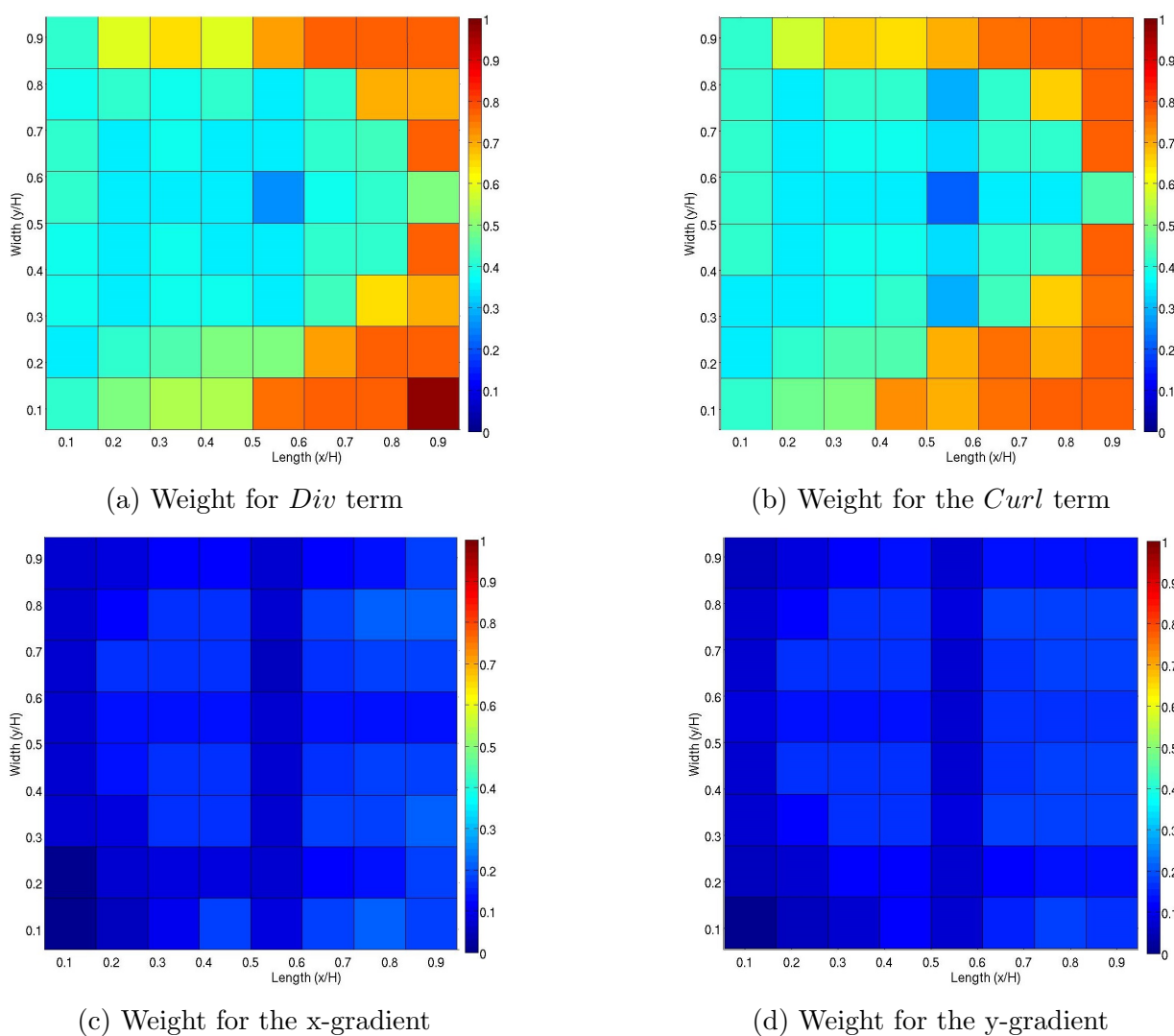


Figure 5.3: Sensitivity map of individual functionals

The sensitivity map has been normalized to the maximum value from all 4. The lower limit of the legend has been changed to the lowest value among the 4. This map indicates the change in the number of iterations or rather the condition of the matrix with perturbations in weight function.

#### 5.1.1.2 Optimization

After obtaining the sensitivity map of the domain and the weight functions, only those elements with most sensitivity were sorted. Among this set, the number of elements provided by equation (4.1) were split. An optimization study was conducted using MATLAB's "fmincon" library. The linear equalities set here normalizes the weights to 1. The optimizer was run for an initial equal weights of [0.1 0.3 0.5 0.1].

Weights	Values
<i>Div</i> term	0.25
<i>Curl</i> term	0.25
x-gradient	0.25
y-gradient	0.25
<b>Number of Iterations</b>	<b>295</b>

Table 5.1: Initial equal weights

Weights	Values
<i>Div</i> term	0.4897
<i>Curl</i> term	0.2744
x-gradient	0.1777
y-gradient	0.0683
<b>Number of Iterations</b>	<b>235</b>

Table 5.2: Optimized weights

The optimized weight was found after 7 outer loop iterations and 45 inner loop iterations. The changes in the outer loop are represented in appendix

### 5.1.2 Case 2:

Consider the same example of a  $10 \times 10$  uniform square grid as shown in the Figure (5.3) below. The difference here is the boundary condition specified.

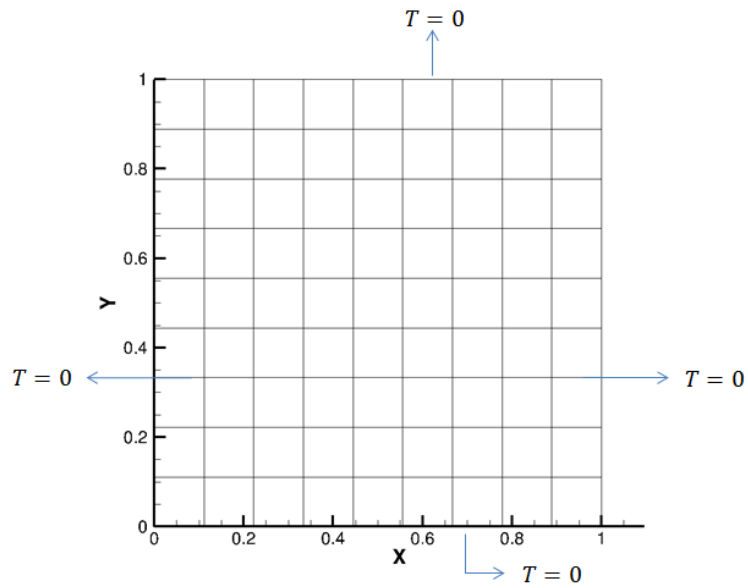
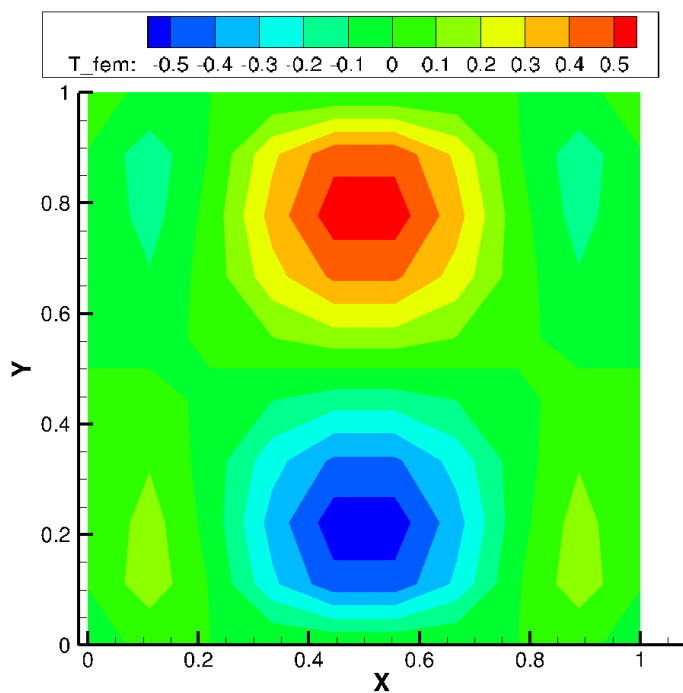


Figure 5.4: A  $5 \times 5$  uniform square grid with constant  $T$  on the boundaries

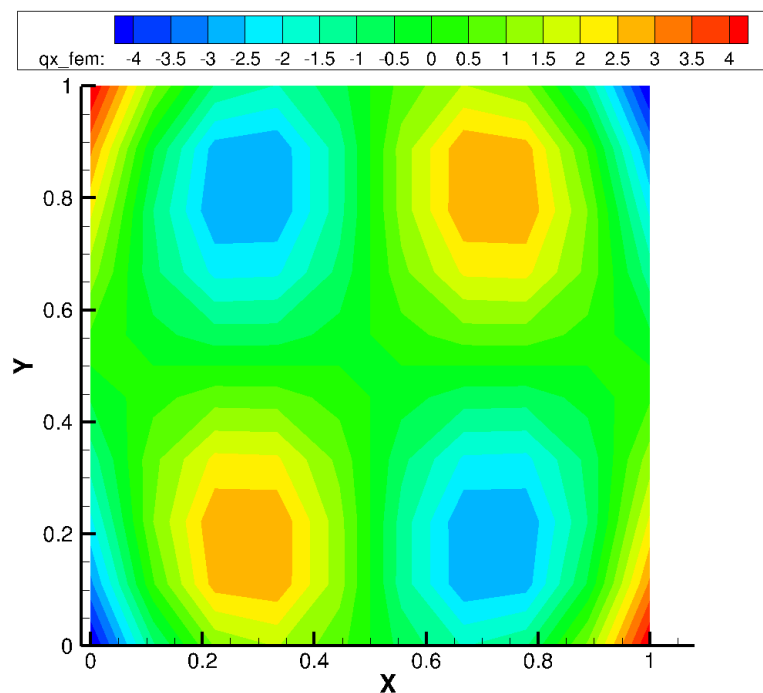
The source term was kept the same from [3],

$$F = (5\pi^2 + 1)\cos(2\pi x)\cos(\pi y) \quad (5.16)$$

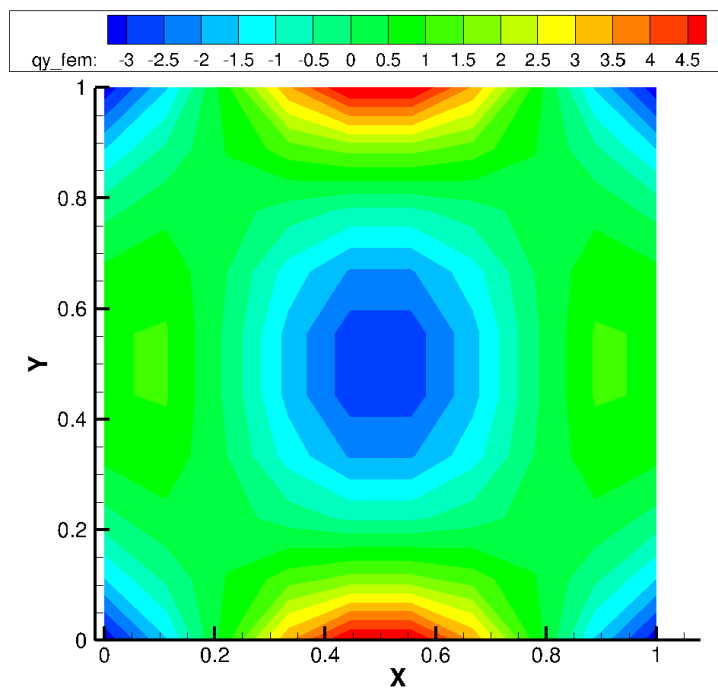
The plot below indicates the least-squares solution of  $T$ ,  $\vec{q}_x$  and  $\vec{q}_y$ .



(a) LSFEM solution for  $T$



(b) LSFEM solution for  $q_x$



(c) LSFEM solution for  $q_y$

Figure 5.5: The plot of LSFEM solution with constant  $T$  on boundaries

### 5.1.2.1 Sensitivity Analysis

Like discussed before, a sensitivity analysis was conducted on an element by element basis. The location to split the quadrilateral elements can be estimated from the sensitivity map. The weight factors were initially normalized to 1 ( $weights = [0.25 \ 0.25 \ 0.25 \ 0.25]$ ). Each weight was perturbed by  $10^{-6}$  and the number of iterations for each case was mapped and has been shown below.

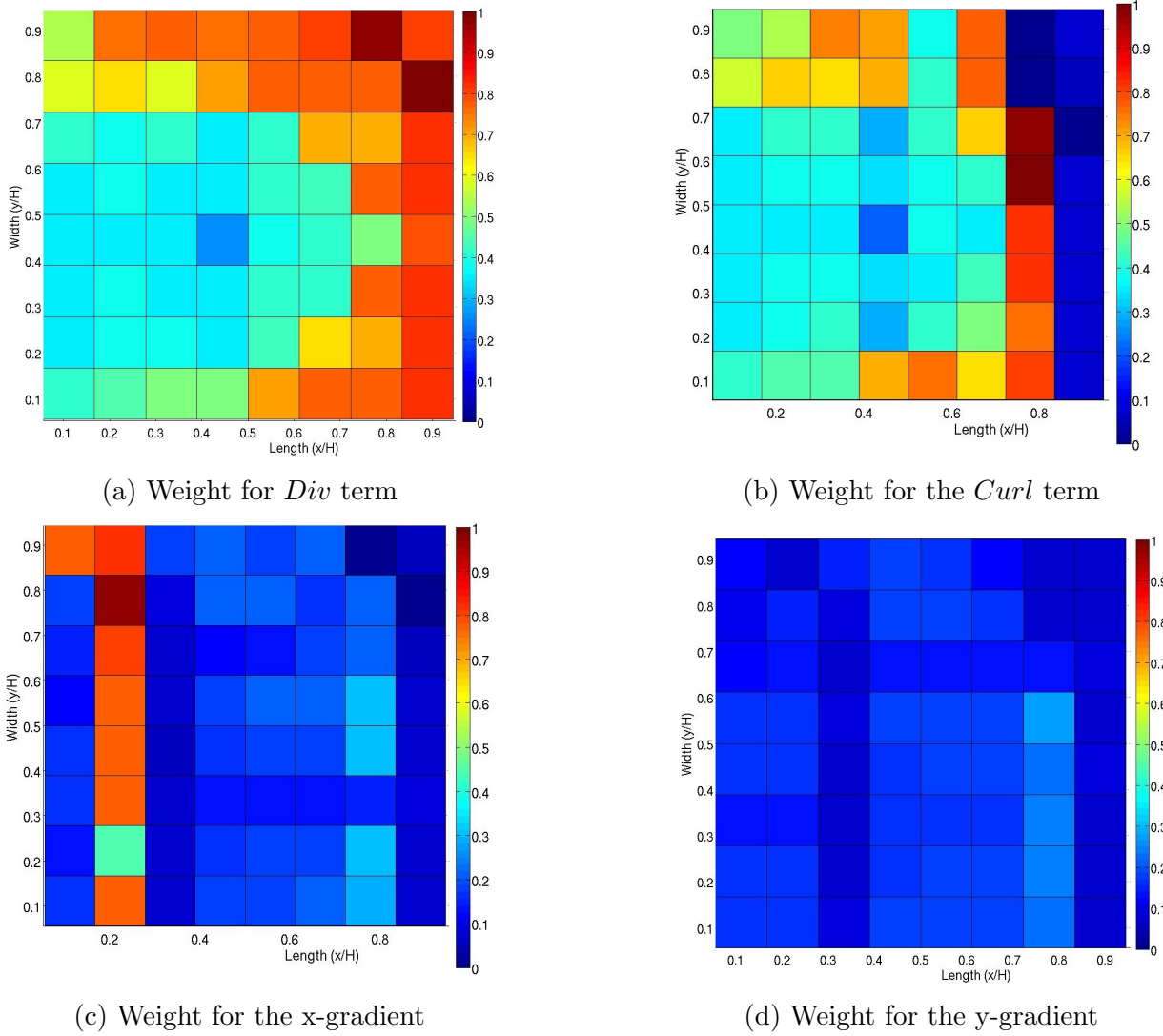


Figure 5.6: Sensitivity map of individual functionals

The sensitivity map has been normalized to the maximum value from all 4. The lower limit of the legend has been changed to the lowest value among the 4. This map indicates the change in the number of iterations or rather the condition of the matrix with perturbations in weight function.

### 5.1.2.2 Optimization

After obtaining the sensitivity map of the domain and the weight functions, only those elements with most sensitivity were sorted. Among this set, the number of elements provided by equation (4.1) were split. An optimization study was conducted using MATLAB's "fmincon" library. The linear equalities set here normalizes the weights to 1. The optimizer was run for an initial equal weights of [0.1 0.3 0.5 0.1].

Weights	Values
<i>Div</i> term	0.25
<i>Curl</i> term	0.25
x-gradient	0.25
y-gradient	0.25
<b>Number of Iterations</b>	<b>246</b>

Table 5.3: Initial equal weights

Weights	Values
<i>Div</i> term	0.4897
<i>Curl</i> term	0.2744
x-gradient	0.1777
y-gradient	0.0683
<b>Number of Iterations</b>	<b>205</b>

Table 5.4: Optimized weights

The optimized weight was found after 7 outer loop iterations and 45 inner loop iterations. The changes in the outer loop are represented in appendix

## 5.2 Convection-Diffusion Equation

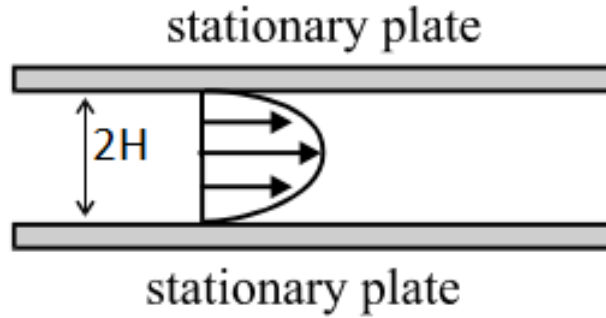


Figure 5.7: Flow through parallel plates

Considering a steady and hydrodynamically fully developed flow between two impermeable parallel plates such that they are  $2H$  apart. The momentum equation can be derived using the Brinkman momentum equation,

$$\frac{\vec{u}}{U_{avg}} = \frac{3}{2} \left[ 1 - \left( \frac{y}{H} \right)^2 \right] \quad (5.17)$$

Convection between 2 parallel plates essentially solves for the scalar variable  $T$  in the Convection-Diffusion equation. This study considers a 2-D steady incompressible Convection-Diffusion equation,

$$\vec{u} \cdot \nabla T - \frac{1}{Pe} \nabla^2 T = 0 \quad (5.18)$$

Equation (5.18) is a non-dimensionalized form of the Convection-Diffusion equation where  $T$  is any scalar variable,  $Pe$  denotes the Peclet Number which represents



the ratio of convection to diffusion,  $\vec{u}$  denotes the velocity. The advantages of using LSFEM is to reduce the order of the differential operator to 'L' as shown in equation (3.1). In this 2-D problem, the differential operator is again a second order PDE and it can be reduced to a first order 'L' operator by introducing flux term( $\vec{q}$ ) in the two direction. The reduced system of equations are,

$$\vec{q} = -k \Delta T \quad \text{in } \Omega \quad (5.19)$$

$$\vec{u} \cdot \vec{q} - \frac{1}{Pe} \nabla \cdot \vec{q} = 0 \quad \text{in } \Omega \quad (5.20)$$

$$\nabla \times \vec{q} = 0 \quad \text{in } \Omega \quad (5.21)$$

Where  $\vec{q}$  represents the flux vector in the domain  $\Omega$ . The equations (5.19) - (5.21) represent 4 functionals that have to be collectively minimized in the 2-D least-squares non-dimensionalized sense. These functionals, when represented in least square sense,

$$\pi_1 = \int_{\Omega} \left[ \vec{q}_x + \frac{\partial T}{\partial x} \right]^2 d\Omega \quad (5.22)$$

$$\pi_2 = \int_{\Omega} \left[ \vec{q}_y + \frac{\partial T}{\partial y} \right]^2 d\Omega \quad (5.23)$$

$$\pi_3 = \int_{\Omega} \left[ \vec{u} \cdot \left( \frac{\partial \vec{q}_x}{\partial x} + \frac{\partial \vec{q}_y}{\partial y} \right) - \frac{1}{Pe} \left( \frac{\partial^2 \vec{q}_x}{\partial x^2} + \frac{\partial^2 \vec{q}_y}{\partial y^2} \right) \right]^2 d\Omega \quad (5.24)$$

$$\pi_4 = \int_{\Omega} \left[ \frac{\partial \vec{q}_x}{\partial y} - \frac{\partial \vec{q}_y}{\partial x} \right]^2 d\Omega \quad (5.25)$$

Upon minimizing the residual to 0, it can be reduced to a form,

$$[K]\{U\} = \{F\} \quad (5.26)$$

### 5.2.1 Case 1:

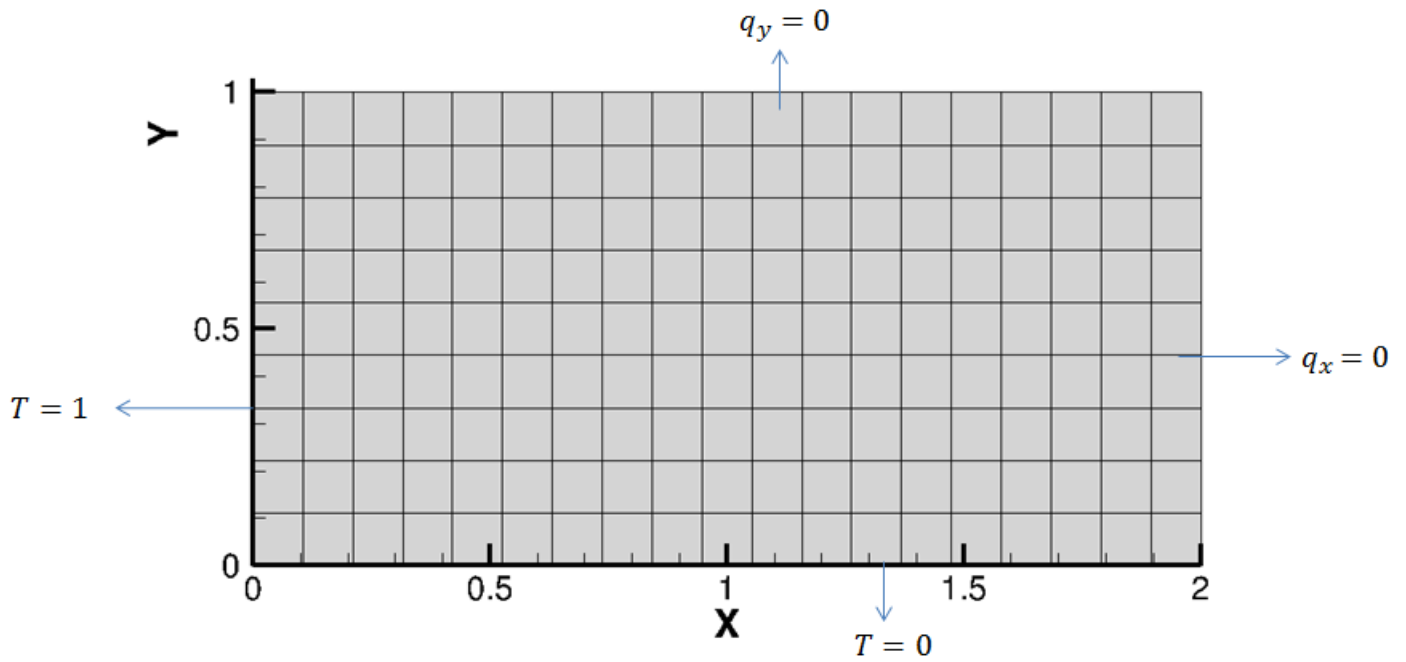
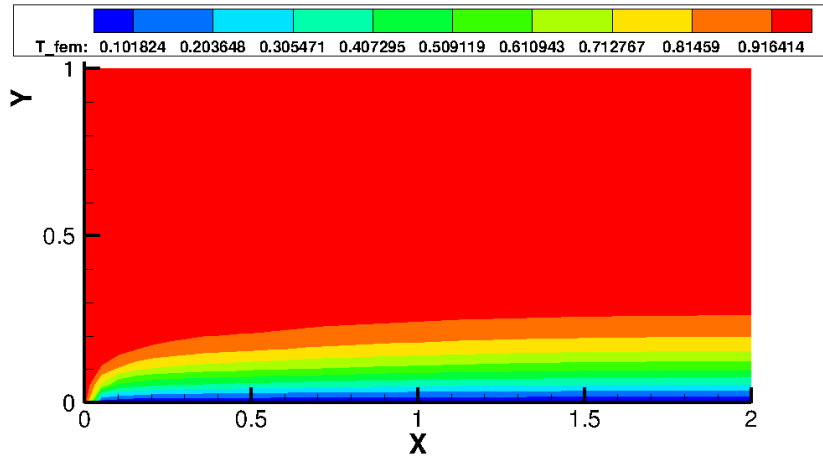


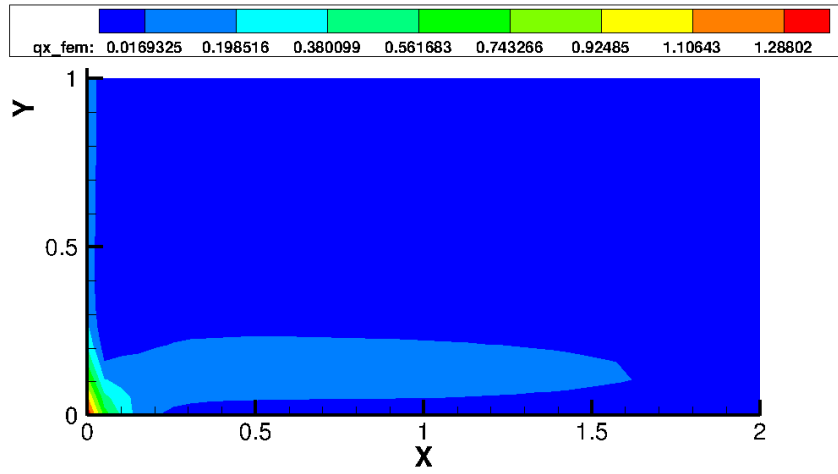
Figure 5.8: Flow through parallel plates with constant wall temperature

Consider a  $20 \times 10$  uniform grid with boundary conditions as shown in figure (5.7). The setup represents only one half of the problem, as symmetry boundary conditions have been used at  $y = 1$ . The length of the domain is taken twice as much as the width.

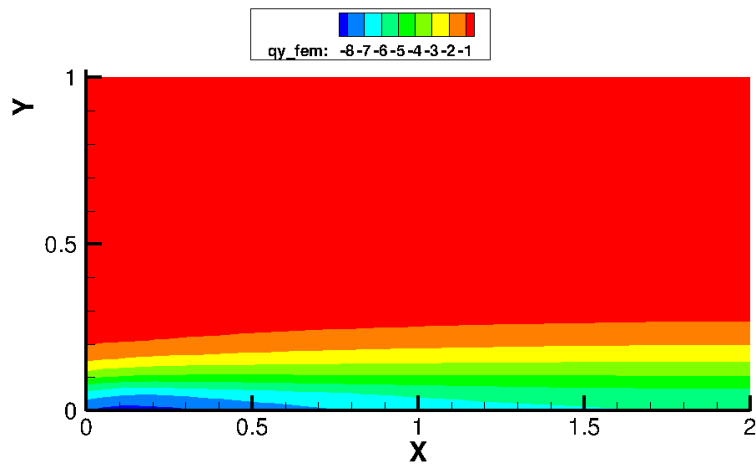
The following simulations have been performed on the same grid and a Peclet number of 5000. This indicates that the flow is a little convection-dominated flow. Cases for more Peclet number have been included in the Appendix A.



(a) LSFEM solution for  $T$



(b) LSFEM solution for  $q_x$



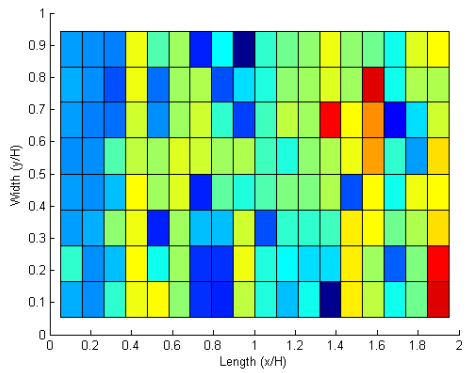
(c) LSFEM solution for  $q_y$

Figure 5.9: The plot of LSFEM solution with constant wall temperature

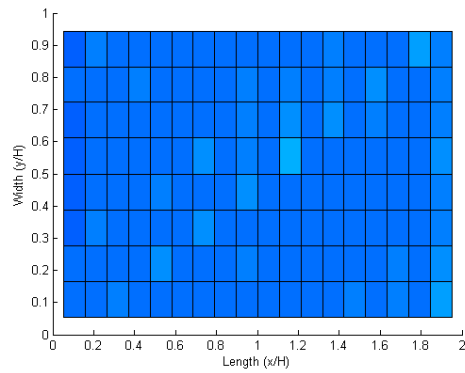
### 5.2.1.1 Sensitivity Analysis

The sensitivity map was conducted on this problem as well with the same perturbation of  $10^{-6}$ .

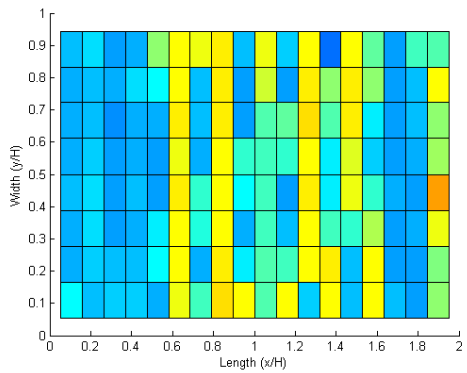
**Peclet No = 1000 :**



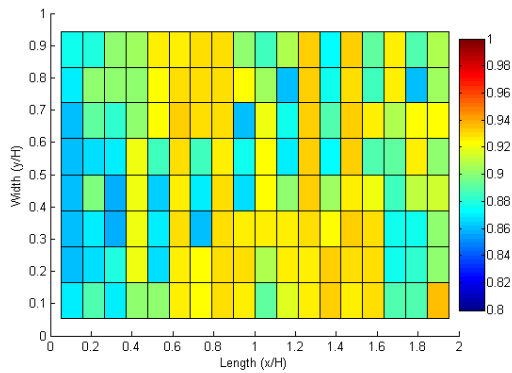
(a) Weight for *Convection - Diffusion* term



(b) Weight for the *Curl* term



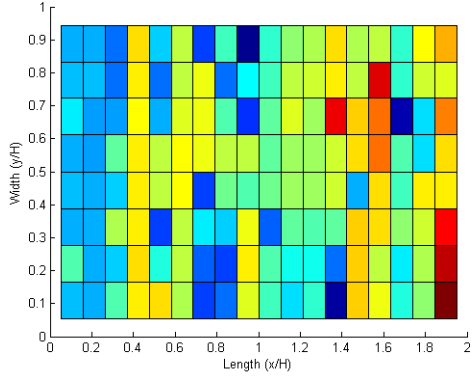
(c) Weight for the *x - gradient*



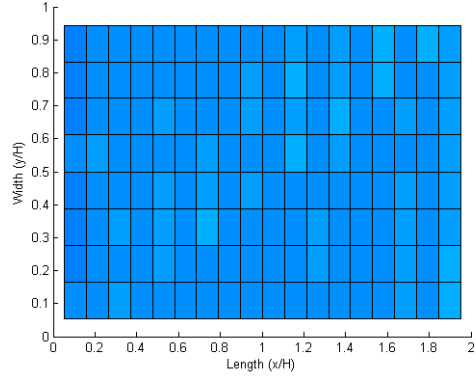
(d) Weight for the *y - gradient*

Figure 5.10: Sensitivity map of individual functionals

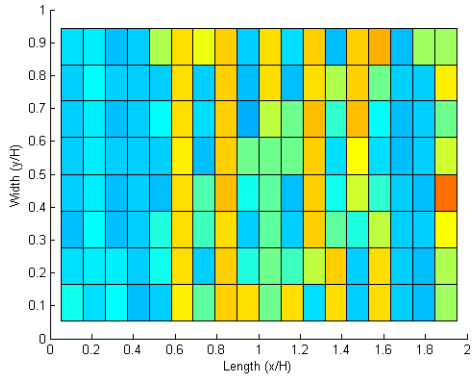
**Peclet No = 10000 :**



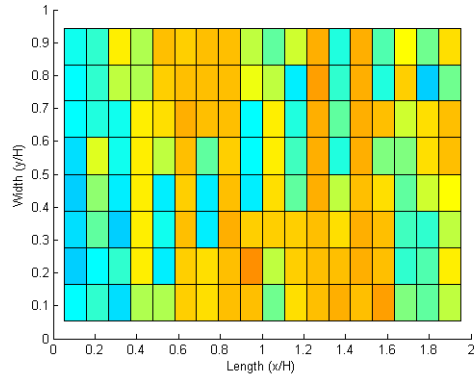
(a) Weight for *Convection - Diffusion* term



(b) Weight for the *Curl* term



(c) Weight for the *x - gradient*



(d) Weight for the *y - gradient*

Figure 5.11: Sensitivity map of individual functionals

### 5.2.1.2 Optimization

After obtaining the sensitivity map of the domain and the weight functions, only those elements with most sensitivity were sorted. Among this set, the number of elements provided by equation (4.1) were split. An optimization study was conducted using MATLAB's "fmincon" library. The linear equalities set here normalizes the weights to 1. The optimizer was run for an initial equal weights of  $[0.1 \ 0.3 \ 0.5 \ 0.1]$ .

Weights	Pe = 500	Pe = 1000	Pe = 10000	Pe = 20000
<i>Con – Dif</i> term	0.25	0.25	0.25	0.25
<i>Curl</i> term	0.25	0.25	0.25	0.25
<i>x-gradient</i>	0.25	0.25	0.25	0.25
<i>y-gradient</i>	0.25	0.25	0.25	0.25
<b>Number of Iterations</b>	<b>331</b>	<b>362</b>	<b>321</b>	<b>318</b>

Table 5.5: Initial equal weights

Weights	Pe = 500	Pe = 1000	Pe = 10000	Pe = 20000
<i>Con – Dif</i> term	0.345	0.3381	0.2939	0.4106
<i>Curl</i> term	0.0544	0.0606	0.1225	0.05
<i>x-gradient</i>	0.3255	0.0837	0.0157	0.2096
<i>y-gradient</i>	0.2852	0.5176	0.4365	0.3396
<b>Number of Iterations</b>	<b>275</b>	<b>288</b>	<b>299</b>	<b>309</b>

Table 5.6: Optimized weights

### 5.2.2 Case 2:

Consider the same  $20 \times 10$  uniform grid with boundary conditions as shown in figure (5.9). The wall boundary is given a constant flux. This problem is solved to spot variations in the least-squares system.

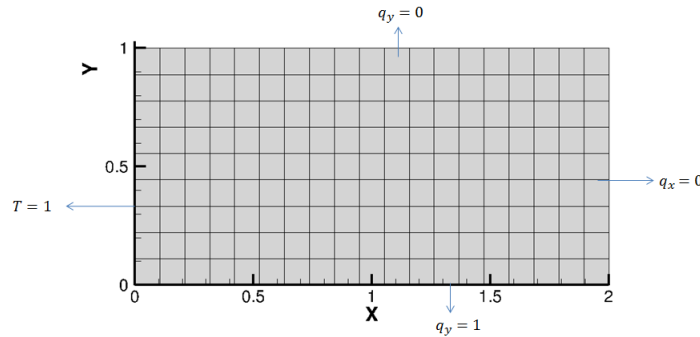
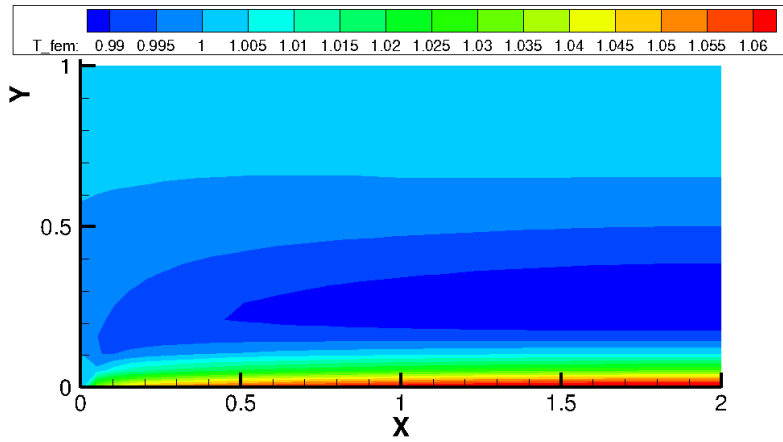
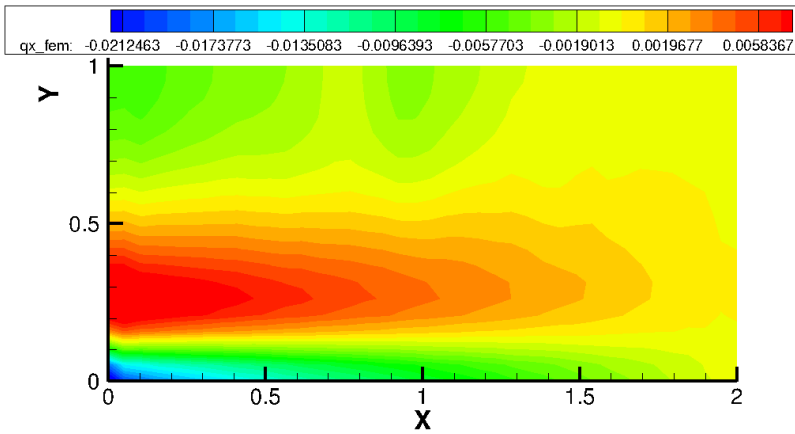


Figure 5.12: Flow through parallel plates with constant flux at the plate boundary

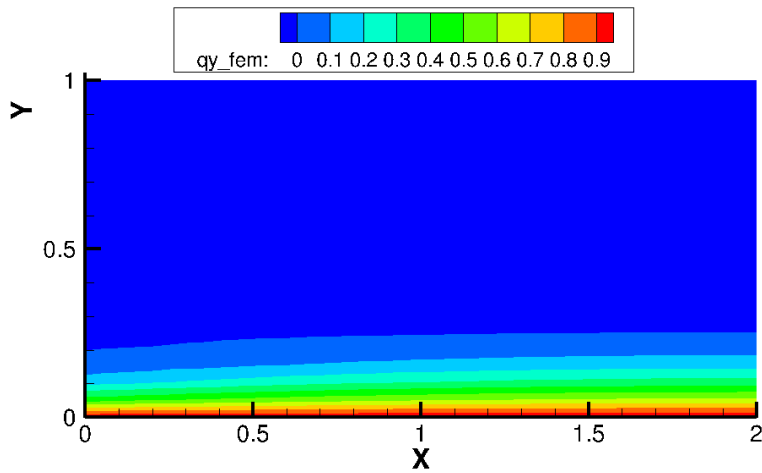
The following simulations have been performed on the same grid and a Peclet number of 5000.



(a) LSFEM solution for  $T$



(b) LSFEM solution for  $q_x$



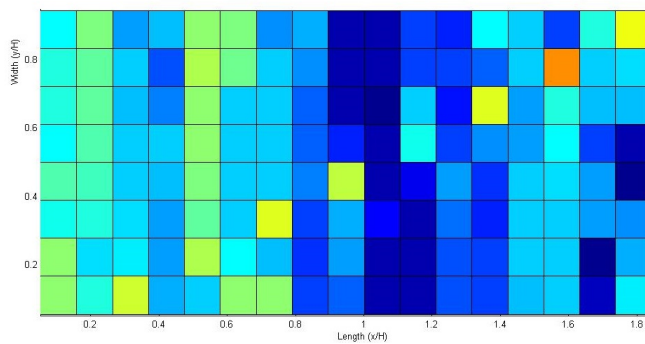
(c) LSFEM solution for  $q_y$

Figure 5.13: The plot of LSFEM solution with constant wall flux

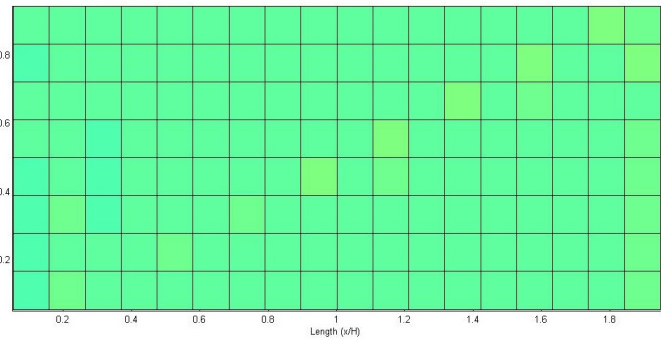
### 5.2.2.1 Sensitivity Analysis

The sensitivity map was conducted on this problem as well with the same perturbation of  $10^6$ .

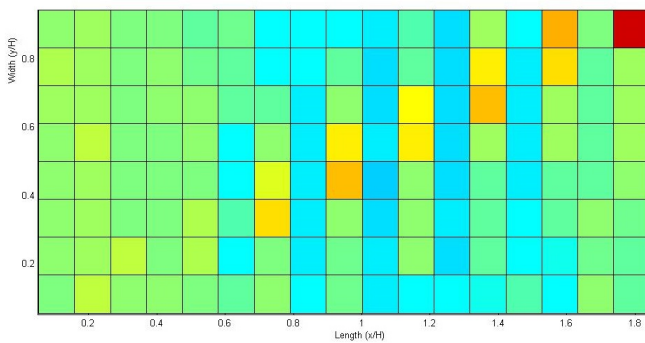
**Peclet No = 1000 :**



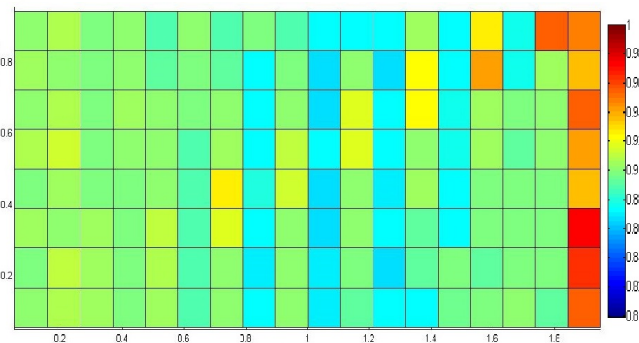
(a) Weight for *Convection - Diffusion* term



(b) Weight for the *Curl* term



(c) Weight for the *x - gradient*



(d) Weight for the *y - gradient*

Figure 5.14: Sensitivity map of individual functionals



Peclet No = 10000 :

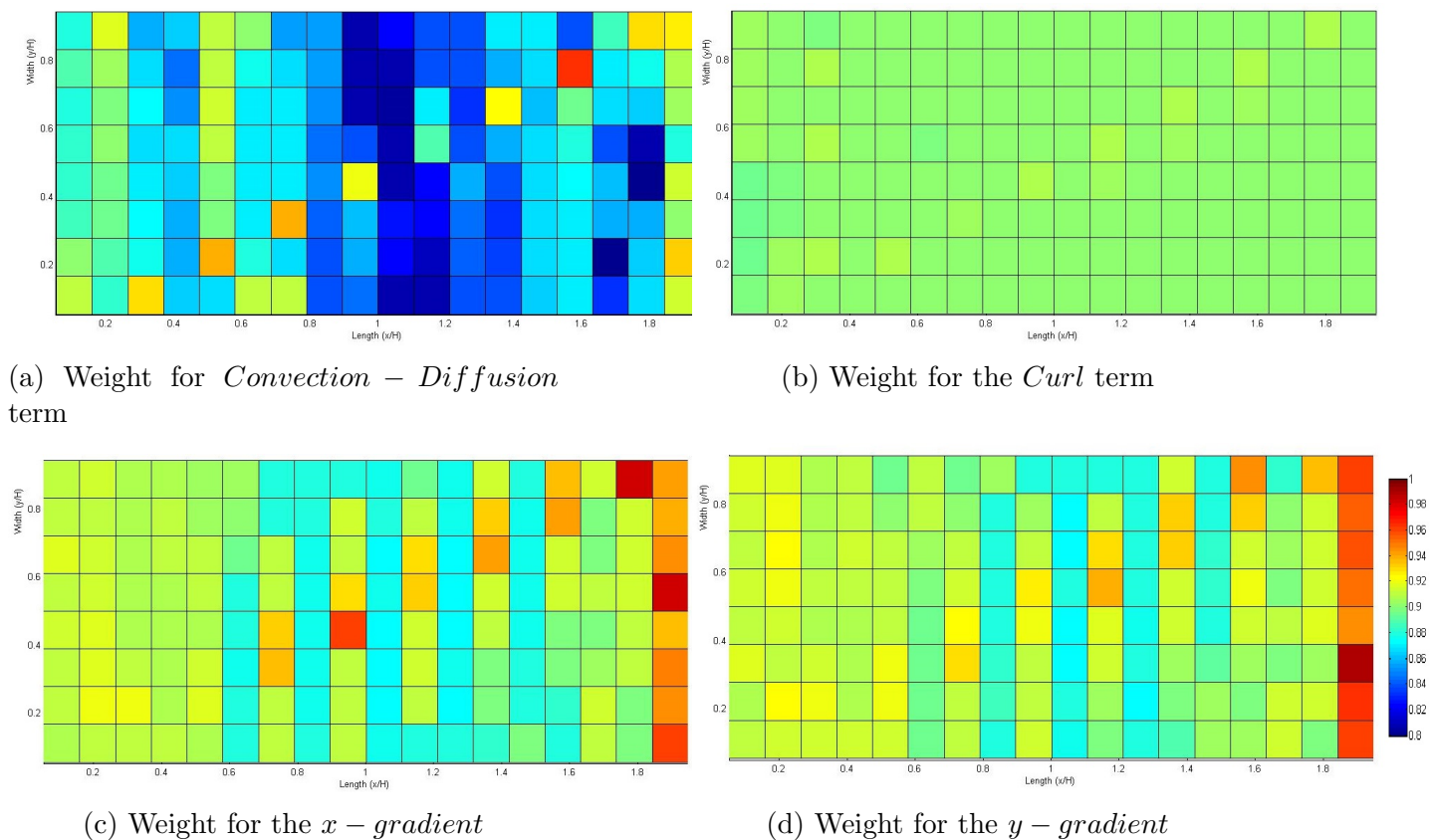


Figure 5.15: Sensitivity map of individual functionals

### 5.2.2.2 Optimization

The linear equalities set here normalizes the weights to 1. The optimizer was run for an initial equal weights of [0.1 0.3 0.5 0.1].

Weights	Pe = 500	Pe = 1000	Pe = 10000	Pe = 20000
<i>Con – Dif</i> term	0.25	0.25	0.25	0.25
<i>Curl</i> term	0.25	0.25	0.25	0.25
<i>x-gradient</i>	0.25	0.25	0.25	0.25
<i>y-gradient</i>	0.25	0.25	0.25	0.25
<b>Number of Iterations</b>	<b>306</b>	<b>335</b>	<b>310</b>	<b>352</b>

Table 5.7: Initial equal weights

Weights	Pe = 500	Pe = 1000	Pe = 10000	Pe = 20000
<i>Con – Dif</i> term	0.345	0.3381	0.2939	0.4106
<i>Curl</i> term	0.0544	0.0606	0.1225	0.05
<i>x-gradient</i>	0.3255	0.0837	0.0157	0.2096
<i>y-gradient</i>	0.2852	0.5176	0.4365	0.3396
<b>Number of Iterations</b>	<b>239</b>	<b>245</b>	<b>270</b>	<b>311</b>

Table 5.8: Optimized weights

## CHAPTER 6

### Conclusion and Future Work

#### 6.1 Conclusion

The least squares finite element has been criticized to have implicit weighting due to the mesh and numerical integration scheme. This leads to a mesh sensitive ill-conditioned problem.

The use of sensitivity analysis technique essentially predicts the more mesh sensitive regions within the domain, where the quadrilateral elements with more Gauss points are split into triangular elements with lower Gauss points, thereby reducing the order of integration. This effectively matches the number of equations to the number of unknowns.

The sensitivity analysis, as the result shows is good map to perform the split. Upon doing the split, the next step is to generate an optimized set of weights that tackles the ill-conditionality problem of certain system. This study has also successfully worked on both elliptic and elliptic-hyperbolic equations. The solution to these equations are independent of weights applied on the functionals.

#### 6.2 Future Work

1. The same methodology can be implemented on a more convection dominated flows. The convective part is the major contributor of the ill-conditionness.
2. It is very important to produce better results with a more global optimizer like say, Genetic Algorithm, where the global minima is the goal.

3. Another extension of this research would definitely be using the same technique on other traditional finite element methods.
4. This study shows that a constant set of weights have been used for all elements. This can be changed to make the weight as a function of the spatial dimensions. Hence, each element would have a different set of weights according to its location. A trend on what kind of function to be used for a particular type of problem can be a parametric study.
5. Non-linearity in the formulations can cause additional ill-conditioning and it is very common while solving Navier-Stokes equation. The condition number for such problems can be better quantified.

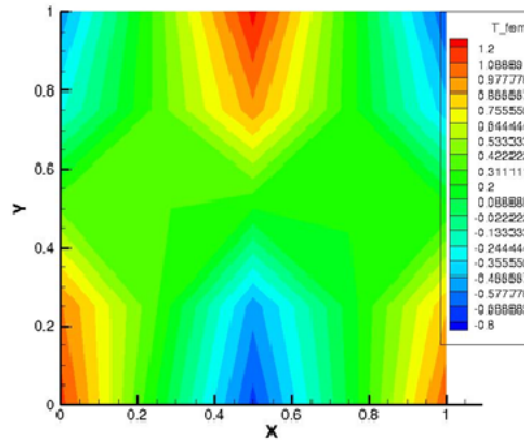
## APPENDIX A

### Validation

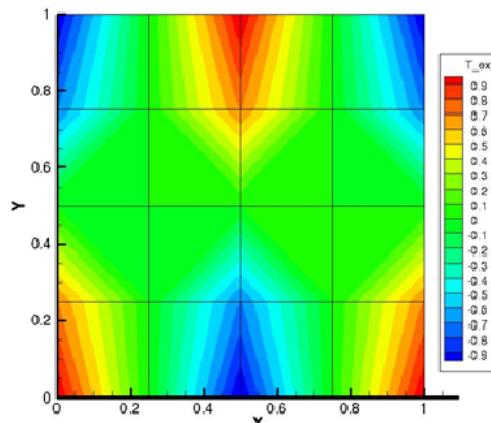
For the validation of results, a  $5 \times 5$  Grid has been considered. The analytical solution is obtained from the method of manufacturing from Jiang's work [3]. The forcing function taken here is  $F = (5\pi^2 + 1)\cos(2\pi x)\cos(\pi y)$ .

A.0.1 For Primary Variable  $T$  :

$$T = -\cos(2\pi x)\cos(\pi y) \tag{A.1}$$



(a) LSFEM solution for  $T$

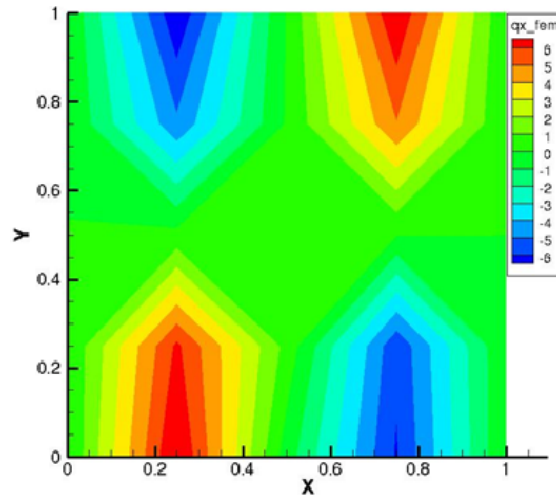


(b) Exact Solution for  $T$

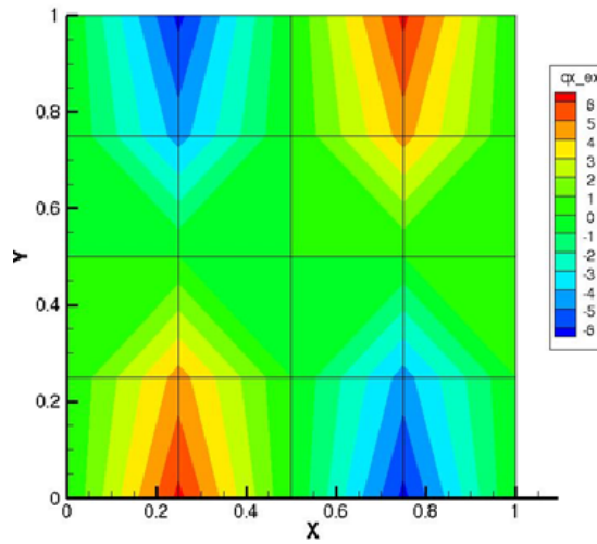
Figure A.1: Comparison of approximate to exact solution for variable  $T$

A.0.2 For Flux Variable  $q_x$  :

$$q_x = 2\pi \sin(2\pi x) \cos(\pi y) \tag{A.2}$$



(a) LSFEM solution for  $q_x$

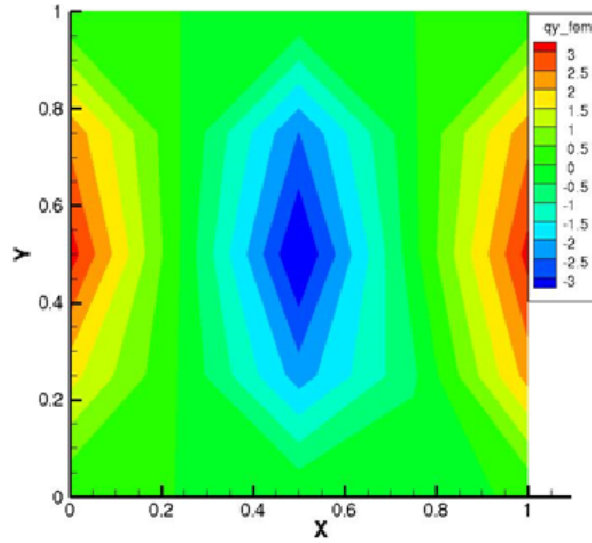


(b) Exact Solution for  $q_x$

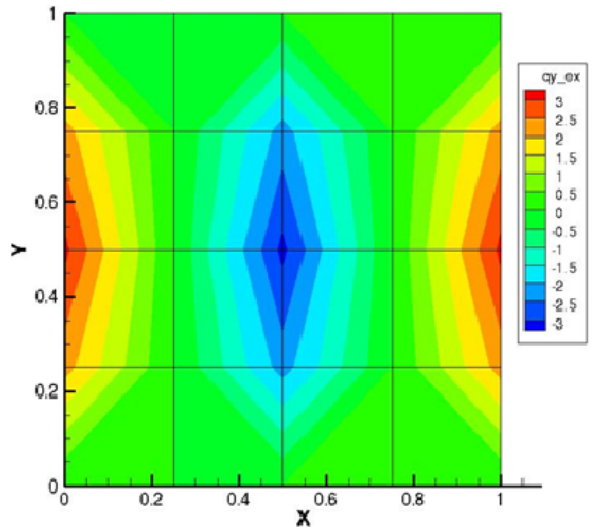
Figure A.2: Comparison of approximate to exact solution for variable  $q_x$

A.0.3 For Flux Variable  $q_y$  :

$$q_y = \pi \cos(2\pi x) \sin(\pi y) \tag{A.3}$$



(a) LSFEM solution for  $q_y$



(b) Exact Solution for  $T$

Figure A.3: Comparison of approximate to exact solution for variable  $q_y$



## APPENDIX B

### Optimizer

The optimizer used here is the '*fmincon*' function in MATLAB. This is a local optimizer that estimates the minimum of constrained non-linear multivariable function.

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ c_{eq}(x) = 0 \\ A.x \leq b \\ A_{eq}.x = b_{eq} \\ l_{bound} \leq x \leq u_{bound}, \end{cases}$$

## REFERENCES

- [1] B.-N. Jiang, “A least-squares finite element method for incompressible Navier-Stokes problems,” *International Journal for Numerical Methods in Fluids*, vol. 14, no. 7, pp. 843–859, 1992.
- [2] B.-N. Jiang and L. a. Povinelli, “Least-squares finite element method for fluid dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 81, no. 1, pp. 13–37, 1990.
- [3] B.-n. Jiang and L. A. Povinelli, “Optimal least-squares finite element method for elliptic problems,” vol. 102, pp. 199–212, 1993.
- [4] Y. Saad, “Parallel iterative methods for sparse linear systems,” *Studies in Computational Mathematics*, vol. 8, no. C, pp. 423–440, 2001.
- [5] E. D. Eason, “A review of least-squares methods for solving partial differential equations,” *International Journal for Numerical Methods in Engineering*, vol. 10, no. 5, pp. 1021–1046, 1976.
- [6] J. Bramble and A. Schatz, “On the numerical solution of elliptic boundary value problems by least squares approximation of the data,” in *Numerical Solution of Partial Differential EquationsII*, 1971, pp. 107–131.
- [7] R. S. Varga, *Least Squares Methods*, pp. 43–49.
- [8] P. P. Lynn and S. K. Arya, “Use of the least squares criterion in the finite element formulation,” *International Journal for Numerical Methods in Engineering*, vol. 6, no. 1, pp. 75–88, 1973.
- [9] Zienkiewicz and Taylor, “Finite Element Method - The Basis (Volume 1),” *Academy of Engineering Polish Academy of Science Chinese Academy of Sciences*

- National Academy of Science Italy (Accademia dei Lincei)*, vol. 35, no. 4, pp. 110–114, 1981.
- [10] P. P. Lynn, “Least squares finite element analysis of laminar boundary layer flows,” *International Journal for Numerical Methods in Engineering*, vol. 8, no. 4, pp. 865–876, 1974.
- [11] P. P. Lynn and K. Alani, “Efficient least squares finite elements for two-dimensional laminar boundary layer analysis,” *International Journal for Numerical Methods in Engineering*, vol. 10, no. 4, pp. 809–825, 1976.
- [12] J. F. Polk and P. P. Lynn, “A least squares finite element approach to unsteady gas dynamics,” *International Journal for Numerical Methods in Engineering*, vol. 12, no. 1, pp. 1–10, 1978.
- [13] G. J. Fix and M. D. Gunzburger, “On least squares approximations to indefinite problems of the mixed type,” *International Journal for Numerical Methods in Engineering*, vol. 12, no. 3, pp. 453–469, 1978.
- [14] G. J. Fix and M. E. Rose, “A comparative study of finite element and finite difference methods for Cauchy-Riemann type equations,” *SIAM J. Numer. Anal.*, vol. 22, no. 2, pp. 250–261, 1985.
- [15] W. L. Wendland, “Review: W. L. Wendland, Elliptic systems in the plane,” *Bull. Amer. Math. Soc. (N.S.)*, vol. 2, no. 1, pp. 203–206, 1980.
- [16] a. K. Aziz, R. B. Kellogg, and a. B. Stephens, “Least squares methods for elliptic systems,” *Mathematics of Computation*, vol. 44, no. 169, pp. 53–70, 1985.
- [17] C. A. Fletcher, “A primitive variable finite element formulation for inviscid, compressible flow,” *Journal of Computational Physics*, vol. 33, no. 3, pp. 301–312, 1979.

- [18] H. Nguyen and J. Reynen, “A space-time least-square finite element scheme for advection-diffusion equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 42, no. 3, pp. 331–342, 1984.
- [19] J. R. Sellars, M. Tribus, and J. S. Klein, “Heat transfer to laminar 110W in a round tube or flat conduit—the Graetz problem extended,” *Transactions of the ASME*, vol. 78, p. 4414148, 1956.
- [20] A. Haji-Sheikh and K. Vafai, “Analysis of flow and heat transfer in porous media imbedded inside various-shaped ducts,” *International Journal of Heat and Mass Transfer*, vol. 47, no. 8-9, pp. 1889–1905, 2004.
- [21] B.-N. Jiang and G. F. Carey, “Adaptive refinement for least-squares finite elements with element-by-element conjugate gradient solution,” *International Journal for Numerical Methods in Engineering*, vol. 24, no. 3, pp. 569–580, 1987.
- [22] G. F. Carey and B. N. Jiang, “Least-squares finite element method and preconditioned conjugate gradient solution,” *International Journal for Numerical Methods in Engineering*, vol. 24, no. 7, pp. 1283–1296, 1987.
- [23] G. F. Carey and B.-N. Jiang, “Element-by-element linear and nonlinear solution schemes,” *Communications in Applied Numerical Methods*, vol. 2, no. 2, pp. 145–153, 1986.
- [24] G. F. Carey and B. Jiang, “Nonlinear preconditioned conjugate gradient and least-squares finite elements,” no. 3, 1987.
- [25] G. F. Carey and Y. K. Cheung, “Mixed operator problems using least squares finite element collocation,” vol. 22, 1980.
- [26] J. Grcar, “Optimal sensitivity analysis of linear least squares,” *Lawrence Berkeley National Laboratory, Report LBNL- 52434*, vol. 99, 2003.
- [27] P. T. Zhi-Quan Luo, “Perturbation Analysis of a Condition Number for Linear Systems,” vol. 15, no. 2, pp. 636–660, 1994.

## Biography

Anirudh Rajagopal was born in Bengaluru, India, in 1991. He received his B.E. degree in Aeronautical Engineering from Visvesvaraya Technological University, India, in 2013. He went on to complete his M.S. degree from The University of Texas at Arlington in 2016, in Mechanical Engineering. From 2013 - 2014, he worked as a CAD Engineer at CADD Centre India Pvt. Ltd. In 2015, he joined as research student in the Computational Fluid Dynamics laboratory at the University of Texas, Arlington. He had a short internship spell at Exa Corporation where he was required to work on mesh generation tools and projects. He joined the CFD Lab at the university again after the internship to continue working on developing finite element techniques for fluid and heat transfer problems. He has served as a student member at the American Institute of Aeronautics and Astronautics.

If more information regarding the thesis is needed, please contact Anirudh at [anirudh.rajagopal@mavs.uta.edu](mailto:anirudh.rajagopal@mavs.uta.edu)