Efficient Finite Element Simulation of Full-System Elastohydrodynamic Lubrication Problems

by

Sarfraz Ahmed

Submitted in accordance with the requirements for the degree of Doctor of Philosophy.

UNIVERSITY OF LEEDS

The University of Leeds
School of Computing

July 2012

The candidate confirms that the work submitted is his own and that the appropriate credit has been given where reference has been made to the work of others.

Abstract

This thesis is concerned with the efficient numerical solution of problems of elastohydrodynamic lubrication (EHL). Our approach is to consider fully-coupled models in which the governing equations for the lubricating film, the elastic deformation and the force balance are each discretized and solved as a single monolithic nonlinear system of algebraic equations. The main contributions of this work are to propose, implement and analyse a novel, optimal, preconditioner for the Newton linearization of this algebraic system, and to assess the development of efficient finite element meshes through both manual tuning and the use of adaptive mesh refinement based upon a posteriori error estimation and control.

Throughout this work, we employ first order finite element discretizations for both the Reynolds equation (for the lubricant) and for the linear elasticity model on a finite domain. The resulting nonlinear algebraic equations are then solved using a quasi-Newton algorithm. For each linear solve a Krylov subspace method is used and a new blockwise preconditioner is presented which is designed to exploit the specific structure that is present in this class of problem. This preconditioner combines the use of multigrid preconditioning for the elasticity block and a separate, efficient, approximation to precondition the Reynolds block. The solver developed in this work can be distinguished into two variants based upon the use of algebraic and geometric multigrid preconditioning of the elasticity block.

Numerical results are presented both for line and point contact problems to validate the implementations and to allow a comparison of the performance and efficiency of the proposed solution strategies compared to the use of a state-of-the-art sparse direct solver at each Newton step. These results demonstrate that the preconditioned iterative approach is both computationally and memory superior to the sparse direct solver. Most importantly, both the computational and memory costs are seen to grow linearly with the number of unknowns.

A locally adaptive solution scheme is also developed for fully-coupled EHL point contact problems. This automates the refinement process to the regions of the domain which exhibit large error in their solutions. Numerical results are presented which demonstrate the performance and effectiveness of the proposed procedure.

Acknowledgements

All praise and the thanks be to the Almighty, for the strength and His blessing in completing this project.

I would like express my sincere appreciation and deep gratitude to my supervisors, Professor Peter K. Jimack and Dr. Christopher E. Goodyer for giving me invaluable assistance, encouragement, support and guidance throughout. This research project would not have been possible without their help. Thanks a lot to both of them. My deepest gratitude to Prof. P. K. Jimack for matters other than research. Also special thanks to Dr. Mark Walkley for his assistance and valuable comments upon my project. My many thanks to all the members in the scientific computation research group for their help.

I would also like to thank Hazara University and the Higher Education Commission (HEC), Pakistan for their financial support throughout this research at the University of Leeds.

Lastly I wish to express my love and gratitude to my late parents for their support and buck up for progressing in life. I won’t forget my brothers and sisters for their endless love, support and encouragement throughout my life in good and bad times. Many many thanks to them for always being there for me. All this would have been impossible without their encouragement and understanding.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

S. Ahmed, C. E. Goodyer, and P. K. Jimack, “An efficient preconditioned iterative solution of fully-coupled elastohydrodynamic lubrication problems”, *Applied Numerical Mathematics*, 62(5) (2012) 649–663.

S. Ahmed, C. E. Goodyer, and P. K. Jimack, “On the Linear Finite Element Analysis of Fully-Coupled Point Contact Elastohydrodynamic Lubrication Problems”, *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, 226 (2012) 350–361.

All material in these articles is the candidate’s own work under the supervision of the co-authors.

Contents

List of Figures

List of Tables

List of Algorithms

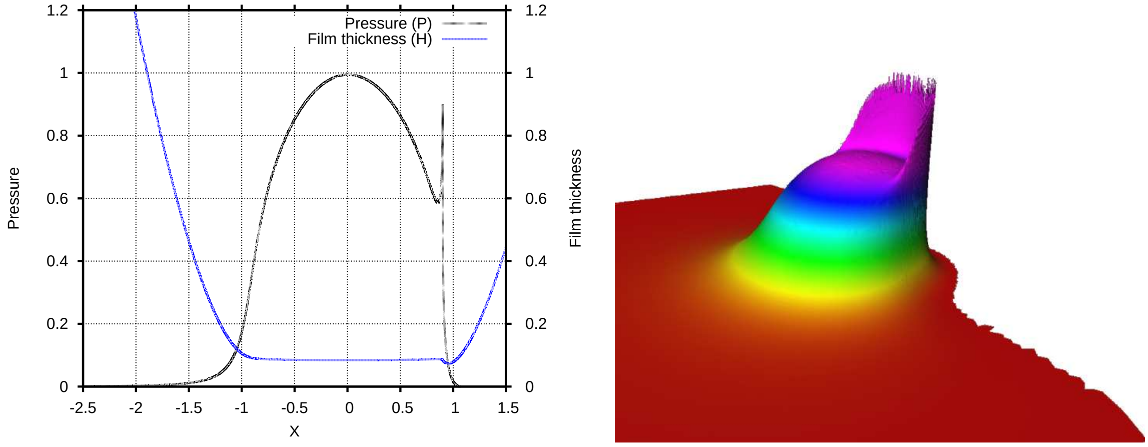
Chapter 1

Introduction

1.1 Elastohydrodynamic Lubrication

Friction is force resisting the motion of objects relative to each other. Although this is essential in many daily life activities such as walking, brushing teeth, stopping moving objects, etc, there are examples where frictional forces are not desirable. Consider the example of mechanical systems, where many individual components are in motion relative to each other. Surfaces will therefore be in contact. Elementary mechanics explains that the frictional force produced not only lowers the efficiency of the components, as work must be done to overcome the friction, but also increases their wear, which affects the life of machine components. Therefore, in such situations, frictional forces need to be minimized in order to obtain maximum efficiency and to prevent wear. A common way to reduce the frictional force and prevent wear is lubrication. The separation of the components by the lubricant helps to protect them from direct contact and hence reduces friction and wear, which not only leads to less energy consumption but also increases the life of components. The use of a lubricant reduces the friction to about a tenth that in a dry contact [102]. To maintain such a lubricant film between the contact surfaces, a pressure is generated inside the lubricant film through the relative motion of the surfaces: this is referred as hydrodynamic lubrication [54, 104].

Elastohydrodynamic lubrication (EHL) deals with the behaviour of a lubricant film between the moving *elastic* components of mechanical systems. The behaviour of the lubricant film is determined by the geometry of the contacting elements. In particular, if the pressure generated inside the lubricant film is sufficiently high (up to giga pascals in the case of steel) then the contacting elements deform elastically and hence define a new shape of the lubricant film. Figure 1.1 illustrates two typical EHL pressure profiles, showing the high pressure regions generated in EHL contacts. It is high pressure, and the resulting elastic deformation, that characterizes elastohydrodynamic lubrication [33, 102, 104]. In addition, with such a wide range of operating pressures in the contact, the properties of lubricants such as viscosity, and density etc., will change across the contact. Thus the fluid film formation is also strongly affected by this lubricant behaviour which can not be neglected.



(a) Line contact: Pressure and film thickness profiles (b) Point contact: Pressure profile

Figure 1.1: Typical EHL solutions - note the typical pressure spike/ridge which occurs on the outflow side of the contact.

In general, contact types are classified into two categories: conforming contacts and non-conforming contacts. In the former case the surfaces of the contacting elements fit exactly or closely together before any deformation takes place. Journal bearings are examples of such a contact. In this case, contact surfaces meet at many points in the undeformed case i.e the contact area is usually large. A non-conforming contact is formed if the contact surfaces meet along a line or at a point prior to any deformation (these are cases (a) and (b) respectively in Figure 1.1). Ball bearings and roller bearings are examples of such contacts. In the non-conforming contacts the contact area is very small compared to size of the contacting elements and stresses are highly concentrated in this region. In other words, for a given load, the pressure generated is very high and therefore may lead to significant elastic deformation of contact surfaces. Such contacts usually fall into elastohydrodynamic lubrication regime [104].

Research into EHL problems comprises of a combination of experiments and numerical simulations. In order to develop efficient oil for the operating conditions for which they are intended, it is essential for designers of lubricants and components to get performance results for a variety of lubricants in different operating conditions. Since applied loads can cause pressure distributions across the contact area of the order of giga-Pascals, minimum film thicknesses are in the micrometre range, and lubricant molecules pass through the contact in hundredths of a second, it can be difficult to conduct physical experiments into the behaviour of EHL contacts. Thus if accurate computer codes are available, then the cost effective solution to numerous EHL tests may be obtained [45]. In short, efficient computer codes thus gain much more significance in tackling these kinds of problem for optimised results, which is the motivation of this study.

1.2 An Historical Overview of Numerical Techniques for EHL Problems

Recall from the previous section that an EHL problem comprises of finding the pressure distribution across the fluid, and the shape of the lubricant film which is determined by the geometry and the resultant elastic deformation of the contacting surfaces. The Reynolds equation [90] governs the pressure distribution across the fluid for a given geometry and the lubricant properties. This is obtained from the Navier-Stokes equations for viscous flow by making a thin film approximation which leads to a nonlinear convection-diffusion equation relating pressure, film thickness and the lubricant properties. The shape of the lubricant film generally depends upon the separation due to undeformed geometry, an initial separation constant, and the elastic deformation of the contacting surfaces. A most commonly used method to calculate the elastic deformation of the contact surfaces is to evaluate an elastic deformation integral [33, 66, 102] which is obtained by an analytical solution of the linear elasticity equation on a semi-infinite domain. For a full description of a complete mathematical model of the EHL problems one is referred to the next chapter.

The history of the numerical solution of EHL problems starts with the work of Petrusevich [85] in 1951, and was first to predict a pressure spike in the outlet region of flow. This provided a platform to the development of different methods. However, circular contact cases were not solved numerically until the early 1970s, see Ranger [88]. The semi-system approach consists in solving the governing EHL equations separately and establishing an iterative procedure between their solutions. One of the first to use this approach were Dowson and Higginson [33] for the line contact case. This followed the pioneering work of Hamrock and Dowson [53] and Ranger [88] for the circular contact.

These methods were based on direct methods, in which the Reynolds equation is solved for the pressure for a given geometry. But the drawback of these methods were their failure for highly loaded cases. To overcome this limitation Ertel [37] introduced the Inverse method. Contrarily to the direct method, this consists of solving Reynolds equation to compute the film thickness for a given pressure profile. Dowson and Higginson [33] were the first to apply this method to EHL line contacts in 1959. This approach was later considered to point contacts by Evans and Snidle [39]. In computational terms, this method was considered undesirable for large systems due to its computational complexity being close to $\mathcal{O}(N^3)$.

A multigrid technique was first used by Lubrecht [73] for both line and point contact cases. This provided a faster convergence rate and hence, reduced the computational cost. Further computation time reduction as achieved by Brandt and Lubrecht [20] introducing a multilevel multi-integration (MLMI) technique, in which computation of the elastic deformation integral was accelerated by reducing its computational complexity using a multilevel framework. Regardless of the efficiency, this approach was found unstable to highly loaded cases. Later on, Venner [102] developed a distributive relaxation scheme for both multilevel techniques to obtain further efficiency, and the stability for highly loaded EHL cases. In the late ‘90s Nurgat [80, 81] presented a simpler new relaxation scheme suitable for highly loaded cases. The convergence of this scheme was further enhanced by Goodyer [45–47]. The author combined this improved technique with adaptive meshing to further the accuracy with least computational work, and improve the efficiency by parallelism. The different works cited here so far are based on finite difference discretizations of the EHL equations. These methods limit the discretization process to regular structured rectangular meshes using low order approximations, and have been the most widely used technique in the EHL modelling.

The finite element method has been applied to EHL problems since the 1970s however. Line contacts were first considered by Taylor and O’Callaghan [99] and point contacts by Oh and Rohde [82]. Recently Lu [70–72] used “discontinuous Galerkin (DG) FEM” to get stabilised results for highly loaded line contact and moderately loaded point contact cases. These DG techniques are not widely used to solve EHL problems today, there is still research being done.

The full-system approach consist in solving the coupled EHL equations simultaneously. A Newton-Raphson full-system approach is first used by Oh and Rohde [82]. This method converges in a few iterations but requires a good initial guess. A similar model was used by Okamura [83], which is improved by Houpert and Hamrock [62] for a line contact case. This model is extended to elliptical contacts by Hsiao et al. [63]. All these methods were based on half space approach for elastic deflection. The drawback of this approach is that it uses the pressure from all points in the domain to calculate the deflection at each point, which makes the resulting linearized system matrix highly dense. Finally, for heavy loads, the Jacobian matrix becomes almost singular which makes it hard to reach the solution.

In 2000, a "differential deflection method" based on the half space approach was introduced by Evans and Hughes [38, 60, 61, 64]. The advantage of this method is to use the information from comparatively fewer points in the domain to calculate the elastic deflection at each point. In other words the influence of pressure acting at a point is reduced to a limited locality of that point. Therefore this approach results in a less dense matrix compared to the half space approach for elastic deflection. Authors and their co-workers applied this method to line contact [64] and then extended to point contact cases [60, 61].

Recently Habchi [49–51] used a numerical approach to compute the solution of classical linear elasticity equation to obtain the elastic deflection. This equation only uses the information at the neighbouring points to calculate the elastic deflection at a point in the domain. So the resultant matrix is highly sparse and made it easy to reach the solution without any special treatment for convergence. The drawback of this method is the need to solve the elasticity equation in a 2D domain for line contact problems and a 3D domain for point contact problems. This cost is minimized by using a fine mesh in the region of interest and a coarse mesh elsewhere. The other advantage of this method is that it yields additional solution information such as displacement, and derived fields such as stress, throughout the solid components, which is not possible using traditional half-space approach. Nevertheless, the relatively high computational cost of this approach has so far prevented its wide spread use.

1.3 Outline of Thesis

In this work the issues of the high computational and memory costs of the fully-coupled approach to solve EHL problems are addressed. Habchi et al. [49–51] used a sparse direct solver to solve the linearized system at each Newton step. Experience shows that sparse direct solvers are very efficient for small systems, but when the resolution and/or the dimension of the problem is increased their performance reduces and they require very large amounts of memory (specifically, both the CPU times and the memory requirements grow significantly faster than $\mathcal{O}(N)$, where N is the number of degrees of freedom). To solve large sparse problems, iterative methods are often considered to be superior to sparse direct methods, however this relies upon finding good preconditioners to make them computationally competitive. In this work an efficient, preconditioned iterative solver has been developed to solve fully-coupled EHL problems. Numerical results presented show that the performance of this technique appears to be close to optimal, both in terms of computational time and memory usage.

In Chapter 2, the mathematical model used for the EHL line and point contact problems is presented. This model can be distinguish into two sets of equations: first the EHL system itself which includes the Reynolds equation, the film thickness equation and the load balance equation. The second set is concerned with the lubricant's properties, such as density and viscosity. For the computation of elastic deflection, two approaches are discussed. The first approach is a traditional half-space approach derived analytically from the linear elasticity equation on a semi-infinite domain. The latter is concerned with the numerical solution of the linear elasticity equation on a finite domain.

In Chapter 3, the various numerical methods that are relevant to this thesis are outlined. This includes a brief introduction to the finite element method. The Newton method is explained for the solution of nonlinear systems. For the solution of linear systems, both direct and iterative approaches are discussed. The use of preconditioning is also explained, where the main focus is given to the multigrid preconditioning (using both geometric and algebraic multigrid). This chapter also explains the implementation of different numerical methods using the KINSOL software library [57].

In Chapter 4, the full-system approach to the solution of EHL problems is discussed. This includes a Galerkin finite element discretization of the EHL line and point contact equations. Since, the Reynolds equation exhibits an oscillatory behaviour in its pressure solution for heavy loads, a Streamline Upwind Petrov-Galerkin (SUPG) method [22] is described to stabilize the pressure solution. A coupling procedure is given which puts all the discrete EHL equations together to form a single, large, nonlinear system. The Newton method is applied to this fully-coupled nonlinear system. For the solution of the linear system formed at each Newton iteration different solution strategies are discussed. For the development of an efficient iterative solver a new blockwise preconditioning strategy is proposed. This preconditioner combines the use of multigrid for the preconditioning of the linear elasticity block and an efficient, separate, approximation to precondition the Reynolds block. Numerical results are presented to justify the accuracy of the line and point contact implementations compared to previously published results.

Chapter 5 is devoted to the solution of EHL line contact problems. The accuracy of the solution is discussed using coarse meshes. The performance of different linear solvers are then discussed to produce a comparison between their computational times and their memory usage. Furthermore, this chapter also describes the effect of varying the Poisson ratio for the elastic material over the accuracy and the performance of the proposed preconditioning strategy. Finally, the performance of the Bi-CGSTAB [101] method is compared with the GMRES method [94].

In Chapter 6, the accuracy and the performance of the solver for EHL point contact problems is considered. First of all, a methodology based on a large number of experiments is explained to select efficient 3D meshes which ensures the accuracy of the elastic deformation solution whilst keeping the EHL solution cost as low as possible. This is then followed by a detailed comparison of the computational times and the memory growth of different linear solvers. Finally, it includes a discussion of the effect of the quality of meshes over the accuracy of the EHL solution.

In Chapter 7, a locally adaptive solution scheme for the fully-coupled EHL point contact problem is discussed. This includes a discussion on an ‘a posteriori’ error estimation, the mesh refinement criteria and the refinement algorithm used. A procedure for post-optimization of the refined meshes is also explained. Finally, numerical results are presented to show the performance and effectiveness of proposed procedure for solving EHL point contact problems.

Finally, the thesis is summarized in Chapter 8. This chapter also describes possible extensions of this research and future work.

Chapter 2

Governing Equations

2.1 Overview

In this chapter the mathematical model used for the EHL problems solved in the rest of this thesis is presented. This model is discussed separately both for the line and the point contact cases and is followed by their non-dimensional forms. This chapter also presents two different groups of dimensionless parameters which significantly reduce the number of physical parameters defining an EHL line or point contact case. Moreover, two different approaches are discussed for computing the elastic deformation of contacting surfaces: the first is the so-called half-space approach [33, 45, 73, 102, 104] which is a mathematical relationship derived from an analytical solution to Lamé's equation of linear elasticity on a semi-infinite domain, while the other is based upon a purely numerical solution of the equation of linear elasticity on a finite domain [50, 51]. The drawback of the latter approach is the need to solve the elasticity equation in a 2D domain for line contact problems and a 3D domain for point contact problems. However, the advantage of this method is that it yields additional solution information such as displacement, and derived fields such as stress, throughout the solid components, which is not possible using the, more traditional, half-space approach (which only yields the surface deformation).

2.2 Geometric Representation of EHL Contacts

Recall from the previous chapter that non-conforming contacts can induce very high pressure (up to giga pascals) in the lubricant film within the contact region and that such wide operating pressure may cause the contact surfaces to deform elastically. Such contacts fall under the EHL regime. These contacts can be categorised into two types: the line contacts and the point contacts. The line contact is formed if the contact surfaces meet along a line prior to any deformation. Figure 2.1 shows a line contact formed between: (a) two parallel cylinders (b) a cylinder and a plane. Here, the x-direction is taken to be perpendicular to the contact and the surfaces have velocities u_1 and u_2 in this direction. The function $h(x)$ presents a measure of the gap between the surfaces. On the other hand, if the surfaces of the contacting elements meet at a point prior to any deformation, then this is referred to as a point contact. Figure 2.2 shows an example of such contact between: two spheres (left) or a sphere and a plane (right).

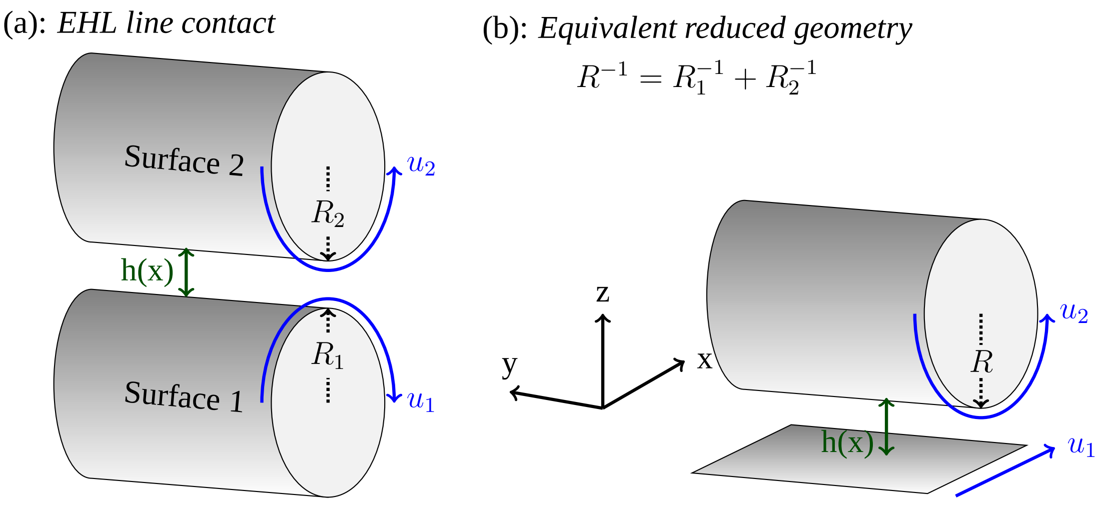


Figure 2.1: An example of a line contact.

Generally the geometry of the contacting machine elements could be rather complex, e.g the contact between gear teeth or the contact between a ball and the inner/outer raceway in a ball bearing. Such geometries can be reduced to a rather more simple form in the immediate vicinity of the contact. The film thickness and the contact area are generally very small compared to overall dimensions of the contacting elements thus the geometry close to contact region can be approximated by paraboloids [33], see Figure 2.3. These parabolically shaped surfaces have local radii of curvature R_{1x} and R_{2x} in the x-direction and R_{1y} and R_{2y} in the y-direction. It can further be reduced to a contact between a plane and an equivalent parabolically shaped surface [33] whose reduced (or equivalent) radii

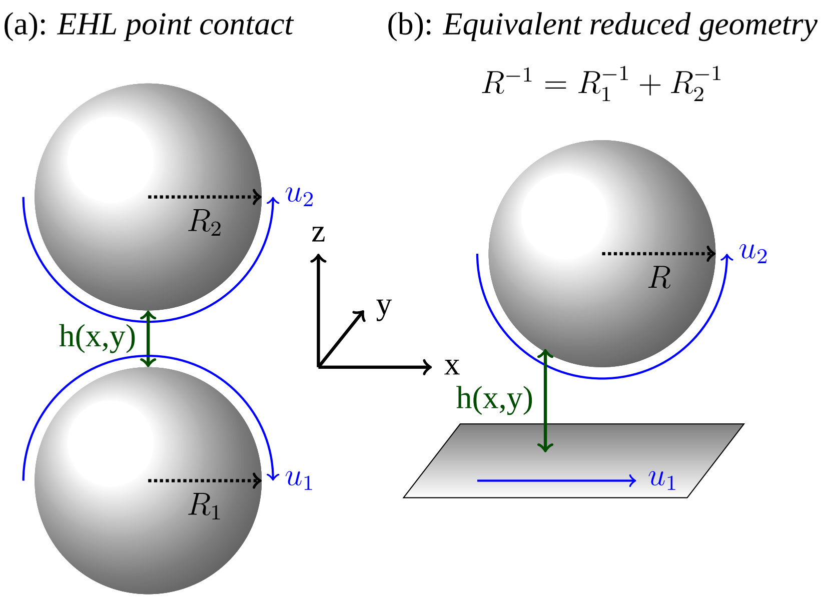


Figure 2.2: An example of a point contact.

of curvatures are:

$$R_x^{-1} = R_{1x}^{-1} + R_{2x}^{-1}$$

$$R_y^{-1} = R_{1y}^{-1} + R_{2y}^{-1}$$

It should be noted that Figure 2.3 can be manipulated as two types of a point contact: a circular point contact if the contacting elements have same radii of curvature in both principal directions otherwise this would be an elliptic point contact [102]. Furthermore, the line contacts assumes an infinitely large radius of curvature of paraboloid in one of the principal directions. Throughout this work, both line contact problems and point contact problems are considered, however in the point contact cases only the circular point contacts have been studied.

2.3 Governing Equations

Models of an EHL contact can be grouped into two sets of equations: one group describes the EHL problem itself while the other is concerned with the properties of the lubricant. The former group consists of the following three equations.

- The Reynolds Equation: this governs the pressure distribution across the contact, for the given geometry and lubricant properties. This is derived from the Navier-Stokes equations for a Newtonian fluid and was first developed by Osborne Reynolds [90] in 1886. For a slow viscous flow, the inertia and body forces are assumed neg-

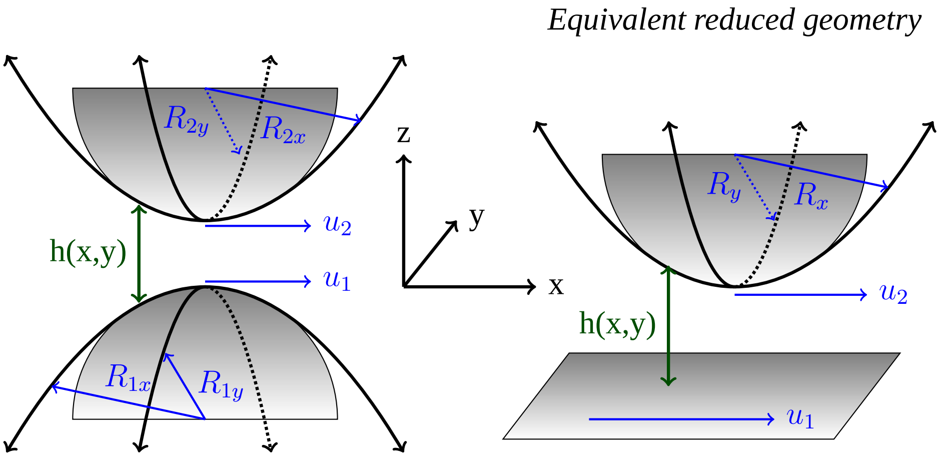


Figure 2.3: Approximation of contact surfaces by paraboloids along with a reduced geometry on the right.

ligible, and therefore neglected, compared to viscous and pressure forces. Another assumption is that flow through a narrow gap eventually leads to a further simplification of these equations of momentum (based upon a thin-film approximation). Using the boundary conditions (based on the assumption that no slip occurs at the boundary surfaces) these equations can be solved for the velocities. Finally substitution of these velocities into the equation of continuity yields an equation for the pressure in the lubricant film called the Reynolds equation.

- The Film Thickness Equation: this determines the shape of the lubricant film across the contact. This is in fact the separation of two surfaces in the contact which generally depends upon a separation constant, the separation due to the undeformed geometry and the elastic deformation of the contacting surfaces induced by the pressure generation within the lubricant film.
- The Force Balance Equation: this is a conservation law which ensures that the total pressure generated inside the lubricant film balances the applied load.

The latter group describes the variation of lubricant properties, such as density and viscosity, with pressure. In the following sections these equations are given separately both for the line and the point contact cases and are discussed in both dimensional and non-dimensional forms.

2.3.1 Line Contact

Recall from previous section that the geometry of contact surfaces in the contact region can be accurately approximated by paraboloids. A further simplification of the contact geometry is obtained by reducing it to an equivalent contact between a paraboloid and a plane. This is known as equivalent or reduced geometry. For the line contact the reduced radius of curvature of the paraboloid approximating the reduced surface is infinitely large in one principal direction (lets say y-direction). Let the two surfaces carrying a lubricant flow in between have the velocities u_1 and u_2 in the x-direction. Let Ω_f denote a domain representing a cross-section through the y-axis (this is ignoring any end effects for the line contact): this gives 1D domain for the lubricant flow and the contact formed. For a given line load, a very high pressure is assumed to be generated in the lubricant film within the contact region. The pressure generated has negligible variation along the y-direction thus the problem reduces to a $1D$ case [33, 104], see Figure 2.4, for example.

Equivalent reduced geometry

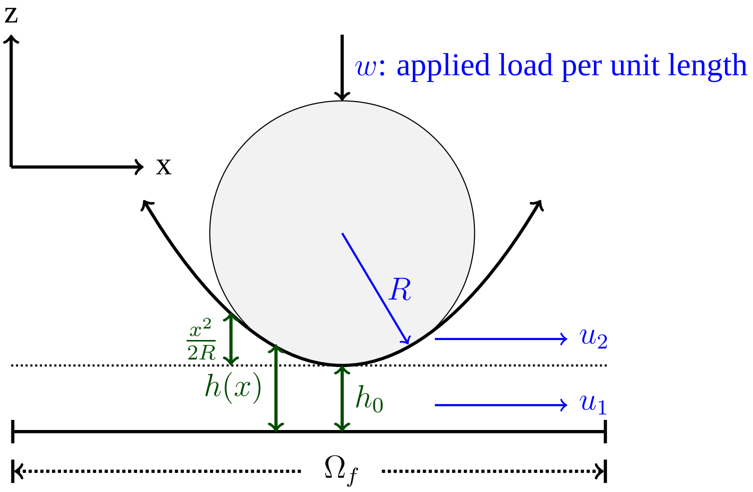


Figure 2.4: A $1D$ line contact problem on the domain Ω_f

To govern the pressure distribution, the Reynolds equation for an isothermal line contact [70] reads:

$$\frac{\partial}{\partial x}\left(\frac{\rho h^3}{\eta}\frac{\partial p}{\partial x}\right) - 6u_s\frac{\partial(\rho h)}{\partial x} - 12\frac{\partial(\rho h)}{\partial t} = 0, \tag{2.1}$$

where

p is pressure,

h is the thickness of the lubricant film,

ρ is the density of the lubricant,

η is the viscosity of the lubricant,

$u_s = u_1 + u_2$ is total velocity of two surfaces with u_1 and u_2 being the individual velocities of upper and lower surfaces respectively.

The film thickness equation for line contact case may be written as:

$$h(x) = h_0 + \frac{x^2}{2R} + d(x), \tag{2.2}$$

where h_0 is the central offset film thickness, the middle term defines the shape of undeformed surface with R being the reduced radius of curvature and $d(x)$ is the combined elastic deformation of the surfaces in contact. This is most commonly calculated using the half-space approximation [33, 66, 102] which is based upon an analytical solution of the linear elasticity equation on a semi-infinite domain, giving:

$$d(x) = -\frac{4}{\pi E'} \int_{-\infty}^{+\infty} \ln|x - x'| p(x') \, \mathrm{d}x',$$

where E' is the reduced elastic modulus of the contacting surfaces given by:

$$\frac{2}{E'} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2}, \tag{2.3}$$

where ν_1 and ν_2 are the Poisson ratios of the materials in the two surfaces.

The conservation law which states that the total pressure generated must be equal to the applied load can be expressed mathematically for the line contact case as:

$$\int_{-\infty}^{+\infty} p(x)\mathrm{d}x = w, \tag{2.4}$$

where w is the applied load per unit length.

2.3.2 Point Contact

In this section we present a mathematical model of EHL point contact problems. In Figure 2.5 a reduced point contact geometry is considered where Ω_f represents the $2D$ fluid domain. The lubricant flow is assumed in the x-direction along with the two surface velocities u_1 and u_2. To govern the pressure distribution in the lubricant film the Reynolds equation for the point contact case reads [45, 70, 104]:

$$\frac{\partial}{\partial x}\left(\frac{\rho h^3}{\eta}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\rho h^3}{\eta}\frac{\partial p}{\partial y}\right) - 6u_s\frac{\partial(\rho h)}{\partial x} - 12\frac{\partial(\rho h)}{\partial t} = 0. \tag{2.5}$$

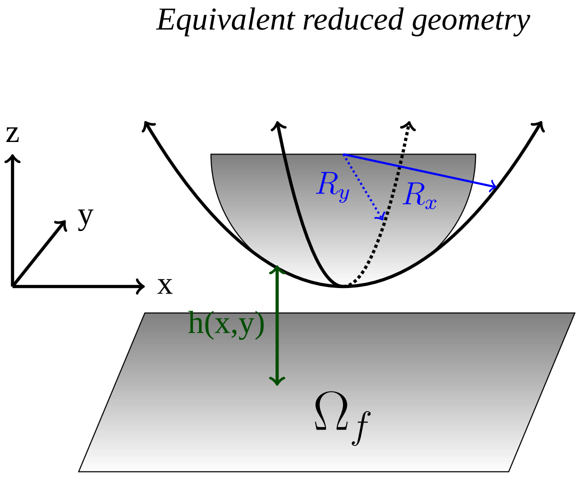


Figure 2.5: A $2D$ point contact problem

The shape of the lubricant film for the point contact case is defined by the following equation:

$$h(x,y) = h_0 + \frac{x^2}{2R_x} + \frac{y^2}{2R_y} + d(x,y), \tag{2.6}$$

where the middle two terms on the right-hand-side give the paraboloid approximation to the shape of an undeformed reduced surface with R_x and R_y being the reduced radii of curvature in x and y directions respectively. For a circular contact R_x = R_y = R, so equation (2.6) becomes:

$$h(x,y) = h_0 + \frac{x^2 + y^2}{2R} + d(x,y), \tag{2.7}$$

where the half-space approximation to the linear elasticity equation [45, 102, 104] gives:

$$d(x,y) = \frac{2}{\pi E'} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{p(x',y')\mathrm{d}x'\mathrm{d}y'}{\sqrt{(x-x')^2 + (y-y')^2}}, \tag{2.8}$$

and E' still satisfies (2.3). Finally, the force balance equation for the point contact case reads:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x,y)\mathrm{d}x\mathrm{d}y = F, \tag{2.9}$$

where F is the applied load.

2.3.3 Lubricant Properties

In EHL contacts there are large variations in pressure, which affect the properties of lubricants across the contact. It is therefore necessary to use lubricant models which can approximate this behaviour. There are two general classes of lubricant model proposed to describe the properties of the oils. The most widely used models are derived purely empirically [11, 33, 91] while the other kind of models aim to be more physically based, e.g. see [10, 59, 107]. The latter models can be much more accurate than those of first kind but still they are rarely used in EHL simulations because of their complex mathematical expressions and dependence upon large numbers of parameters (some of which are hard to determine). On the other hand the empirical models are much more simple, easy to implement and therefore widely used in EHL simulations. In the following only two, commonly used, models are considered. Indeed, these are assumed throughout this thesis both for the line and the point contact problems.

2.3.3.1 Density Model

A commonly used density variation model for EHL problems is that of Dowson and Higginson [33]. It takes account of compressibility of the lubricant and is given by:

$$\rho\left(p\right) \;=\; \rho_0 \frac{0.59\times 10^9 + 1.34p}{0.59\times 10^9 + p}, \tag{2.10}$$

where ρ_0 is density at ambient pressure. This model is only pressure dependent and does not depend on other properties of lubricants. A mathematical equivalent expression seen in [45] is:

$$\rho\left(p\right) \;=\; \rho_0 \left(1 + \frac{0.59\times 10^{-9}p}{1 + 1.7\times 10^{-9}p}\right). \tag{2.11}$$

2.3.3.2 Viscosity Model

The viscosity is another important property of the lubricants in EHL contacts and it may vary over several orders of magnitude with increasing pressure. The simplest viscosity-pressure relationship is exponential, known as the Barus model [11], and is given by:

$$\eta\left(p\right) \;=\; \eta_0 \exp\left(\alpha p\right), \tag{2.12}$$

where η_0 is viscosity at ambient pressure and α is pressure-viscosity coefficient. However, this model is only accurate for relatively low pressures and tends to significantly overestimate the viscosity against high pressure. A more realistic viscosity model was introduced by Roelands [91] which is accurate for pressures up to 1 G Pa and is given by:

$$\eta\left(p\right) \;=\; \eta_0 \exp\left(\frac{\alpha p_0}{z}\left(-1 + \left(1 + \frac{p}{p_0}\right)^z\right)\right), \tag{2.13}$$

where

p_0 is a constant, typically $p_0 = 1.96 \times 10^8$ [Pa],

z is the pressure-viscosity index, typically $0.5 \leq z \leq 0.7$.

Note that the Roelands viscosity model is assumed throughout this thesis.

2.4 Non-dimensionalisation

In EHL problems, the maximum pressure can rise to giga pascals, whilst minimum film thickness may be in the micrometre range, so handling these ranges of numerical values with finite precision arithmetic requires care. In order to numerically compute solutions, and minimise any floating point arithmetic errors, non-dimensionalisation is used. The non-dimensionalisation of EHL equations is mainly based on Hertz's theory for dry contacts [55]. In the following the non-dimensionalisation of the EHL equations is discussed both for the line and the circular point contact cases. Since this thesis is only concerned with steady-state solution methods therefore non-dimensionalisation of such problems are only considered.

2.4.1 Line Contact

The first parameters to be introduced are the maximum Hertzian pressure and the Hertzian radius, which are derived from Hertz's theory [55]. In the case of line contacts, the Hertzian pressure profile is given by:

$$p(x) = \begin{cases} p_h\sqrt{1 - \left(\frac{x}{a}\right)^2} & \text{if } |x| < a \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

where p_h is the maximum Hertzian pressure:

$$p_h = \frac{2w}{\pi a},$$

and the Hertzian half contact width, a, is defined by:

$$a = \sqrt{\frac{8wR}{\pi E'}},$$

where R is the reduced radius of curvature, w is applied load per unit length and E' is the reduced elastic modulus as described earlier in this chapter.

Let us define the following dimensionless variables as:

$$X = \frac{x}{a} \qquad \bar{\rho} = \frac{\rho}{\rho_0} \qquad \bar{\eta} = \frac{\eta}{\eta_0} \qquad P = \frac{p}{p_h}$$

$$H = \frac{hR}{a^2} \qquad D = \frac{d\,R}{a^2}.$$

Hence

$$X = \frac{x}{a} \;\Rightarrow x = aX \;\Rightarrow \mathrm{d}x = a\mathrm{d}X,$$

$$P = \frac{p}{p_h} \;\Rightarrow p = p_h P \;\Rightarrow \mathrm{d}p = p_h \mathrm{d}P.$$

For an isothermal steady state flow, the Reynolds equation (2.1) may be written as:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\rho h^3}{\eta}\frac{\mathrm{d}p}{\mathrm{d}x}\right) - 6u_s\frac{\mathrm{d}(\rho h)}{\mathrm{d}x} = 0, \tag{2.15}$$

using the dimensionless variables, equation (2.15) may be written as:

$$\frac{\mathrm{d}}{\mathrm{d}X}\left(\frac{\bar{\rho}H^3}{\bar{\eta}\lambda}\frac{\mathrm{d}P}{\mathrm{d}X}\right) - \frac{\mathrm{d}}{\mathrm{d}X}\left(\bar{\rho}H\right) = 0, \tag{2.16}$$

where

$$\lambda = \frac{6u_s\eta_0 R^2}{a^3 p_h}.$$

Using the same dimensionless variables, equation (2.2) may be written as:

$$H(X) = H_0 + \frac{X^2}{2} + D(X), \tag{2.17}$$

where

$$D(X) = -\frac{1}{\pi}\int_{-\infty}^{+\infty} \ln|X - X'|P\left(X'\right)dX'. \tag{2.18}$$

Finally, the force balance equation (2.4) becomes:

$$\int_{-\infty}^{+\infty} P(X)\mathrm{d}X = \frac{\pi}{2}. \tag{2.19}$$

In EHL contacts, the number of physical parameters can be significantly reduced into a set of dimensionless parameters. The physical parameters to be reduced are α, E', η_0, R, w, and u_s, and two different families of reduced parameters have been widely used. The first set of parameters to be introduced here is known as Moes parameters [102], M

and L. By defining:

$$\bar{\alpha} \;=\; \alpha p_h$$

and recalling λ from equation (2.16), the Moes parameters, M and L may be defined from:

$$\bar{\alpha} \;=\; L\sqrt{\frac{M}{2\pi}},$$

and

$$\lambda \;=\; \frac{3\pi^2}{8M^2}.$$

Dowson and Higginson [33] introduced a second family of three non-dimensional parameters: the load parameters W, the material parameters G and the speed parameters U to characterize each load situation. These define the following relationships for the line contact:

$$\begin{aligned} W &= \frac{w}{E'R}, \\ G &= \alpha E', \\ U &= \frac{\eta_0 u_s}{2E'R}. \end{aligned}$$

These parameters are related to Moes parameters [102] in the following expressions:

$$M \;=\; W\left(2U\right)^{-1/2},$$

and

$$L \;=\; G\left(2U\right)^{1/4}.$$

These parameters are enough to define any non-dimensional line contact case.

2.4.2 Point Contact

In case of a circular contact, the Hertzian pressure profile is given by:

$$p(x,y) = \begin{cases} p_h\sqrt{1-\left(\frac{x}{a}\right)^2-\left(\frac{y}{a}\right)^2} & \text{if } |x^2+y^2| < a^2 \\ 0 & \text{otherwise} \end{cases} \tag{2.20}$$

where p_h is the maximum Hertzian pressure:

$$p_h \;=\; \frac{3F}{2\pi a^2},$$

and a is the Hertzian radius:

$$a^3 \;=\; \frac{3FR}{2E'},$$

where R is the reduced radius of curvature, F is applied point load and E' is the reduced elastic modulus.

Let us define the following dimensionless variables as:

$$X = \frac{x}{a} \qquad Y = \frac{y}{a} \qquad \bar{\rho} = \frac{\rho}{\rho_0} \qquad \bar{\eta} = \frac{\eta}{\eta_0} \qquad P = \frac{p}{p_h}$$

$$H = \frac{hR}{a^2} \qquad D = \frac{d\,R}{a^2}$$

Hence

$$X = \frac{x}{a} \;\Rightarrow x = aX \;\Rightarrow \partial x = a\partial X,$$

$$Y = \frac{y}{a} \;\Rightarrow y = aY \;\Rightarrow \partial y = a\partial Y,$$

$$P = \frac{p}{p_h} \;\Rightarrow p = p_h P \;\Rightarrow \partial p = p_h \partial P.$$

For an isothermal steady state flow, the Reynolds equation (2.5) may be written as:

$$\frac{\partial}{\partial x}\left(\frac{\rho h^3}{\eta}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\rho h^3}{\eta}\frac{\partial p}{\partial y}\right) - 6u_s\frac{\partial(\rho h)}{\partial x} = 0, \tag{2.21}$$

using the dimensionless variables, equation (2.21) may be written as:

$$\frac{\partial}{\partial X}\left(\frac{\bar{\rho}H^3}{\bar{\eta}\lambda}\frac{\partial P}{\partial X}\right) + \frac{\partial}{\partial Y}\left(\frac{\bar{\rho}H^3}{\bar{\eta}\lambda}\frac{\partial P}{\partial Y}\right) - \frac{\partial}{\partial X}\left(\bar{\rho}H\right) = 0, \tag{2.22}$$

where

$$\lambda = \frac{6u_s\eta_0 R^2}{a^3 p_h}.$$

Using the same dimensionless variables, equation (2.7) may be written as:

$$H = H_0 + \frac{X^2 + Y^2}{2} + D(X, Y), \tag{2.23}$$

where

$$D(X, Y) = \frac{2}{\pi^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{P(X', Y')\mathrm{d}X'\mathrm{d}Y'}{\sqrt{(X - X')^2 + (Y - Y')^2}}. \tag{2.24}$$

Finally, the force balance equation (2.9) becomes:

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} P(X,Y)\mathrm{d}X\mathrm{d}Y = \frac{2\pi}{3}. \tag{2.25}$$

Similar to the line contact case, the parameters:

$$\bar{\alpha} = \alpha p_h$$

and

$$\lambda = \frac{6u_s\eta_0 R^2}{a^3 p_h}$$

are related to Moes parameters [102], M and L, as follows:

$$\bar{\alpha} = \frac{L}{\pi}\left(\frac{3M}{2}\right)^{1/3},$$

and

$$\lambda = \left(\frac{128\pi^3}{3M^4}\right)^{1/3}.$$

In the late 1970s, Hamrock and Dowson [52] introduced the following relations for the point contact problems:

$$\begin{aligned} W &= \frac{F}{E'R^2}, \\ G &= \alpha E', \\ U &= \frac{\eta_0 u_s}{2E'R}. \end{aligned}$$

These parameters are related to Moes parameters [102] in the following expressions:

$$M = W\left(2U\right)^{-3/4},$$

and

$$L = G\left(2U\right)^{1/4}.$$

Again these parameters are enough to define any non-dimensional circular point contact case.

2.4.3 Lubricant Properties

The dimensionless variables:

$$\bar{\rho} = \frac{\rho}{\rho_0} \qquad \bar{\eta} = \frac{\eta}{\eta_0} \qquad P = \frac{p}{p_h}$$

can be used to obtain the dimensionless forms of the density-pressure relationship and the viscosity-pressure relationships defined above. The Dowson and Higginson density model given in equation (2.10) has the following dimensionless form:

$$\bar{\rho}\,(P) \;=\; \frac{0.59 \times 10^9 + 1.34 P\, p_h}{0.59 \times 10^9 + P\, p_h}. \tag{2.26}$$

Also, the dimensionless Roelands viscosity model defined in equation (2.13) is given by:

$$\bar{\eta}\,(P) \;=\; exp\left(\frac{\alpha p_0}{z}\left(-1 + \left(1 + \frac{P\, p_h}{p_0}\right)^z\right)\right). \tag{2.27}$$

2.5 A Modified Reynolds Equation

The solution of differential equations depends upon the boundary conditions. Of the equations discussed so far, the only differential equation to be discussed explicitly is the Reynolds equation, see equations (2.16) and (2.22), which needs to be solved on a 1D fluid domain for line contact problems, and a 2D fluid domain for point contact problems. Let Ω_f represent the fluid domain within which an EHL contact is defined. A specification of pressure P is required at the boundary $\partial\Omega_f$ of fluid domain Ω_f in order to determine the pressure distribution across the fluid domain Ω_f. Generally, it is assumed that, at the boundary of the contact region Ω_f, the pressure of the lubricant is equal to the ambient pressure. Pressure lower than the vapour pressure is physically unacceptable, thus the fluid will cavitate and the pressure will remain equal to the vapour pressure. This process is called cavitation [36, 41, 104], and since both the vapour pressure and the atmospheric pressure are very small compared to the pressure generated inside an EHL contact, they can be treated as zero, hence the pressure is limited from below by zero. Moreover, the principle of mass conservation applied at the boundary of the cavitation region [36] results in the following dimensionless Reynolds boundary conditions:

$$P \geq 0 \;\text{ on }\; \Omega_f \quad \text{and} \quad \nabla P.\vec{n} = 0 \quad \text{at the cavitation boundary,} \tag{2.28}$$

where $\vec{n}$ is the unit outward normal vector to the cavitation boundary. Since the exact location of this boundary is unknown prior to computing the pressure this is known as free boundary problem. Various treatments are possible to handle the free boundary problem, see [36, 41, 105], for example. In this work, a penalty method introduced by Wu [105] is used to handle the cavitation region. This method introduces an additional term (known as the penalty term) for which the Reynolds equation (in both the line and the point contact cases) modifies to

$$\nabla.\left(\epsilon\nabla P\right) - \frac{\partial}{\partial X}\left(\bar{\rho}H\right) - \xi P^{-} = 0, \qquad \text{throughout } \Omega_f, \tag{2.29}$$

where $P = 0$ is imposed on the computational domain boundary $\partial\Omega_f$, ξ a suitably large positive number and $P^{-} = \min(P, 0)$. Note that this additional term has no effect where $P \geq 0$; however, it dominates in the regions where $P < 0$. The term therefore has an effect of forcing the negative pressure towards zero provided that ξ is sufficiently large.

2.6 Linear Elastic Model

In the previous sections the analytic expressions from the half-space approximation to the linear elasticity equation have been used to calculate the elastic deformation of contacting surfaces in semi-infinite elasticity domains. The half-space approximations only provide the elastic deformation at the surfaces of contacting elements and therefore no knowledge is provided as to the full displacement or stress fields within the contacting elements. Moreover the elastic deformation at each point requires the information of overall pressure distribution through the fluid domain Ω_f. In an alternative approach [49–51] the elastic deformation D of the contacting bodies can also be modelled by solving Lamé's equation of linear elasticity numerically on a finite two dimensional domain Ω for line contact problems and a finite three dimensional domain Ω for point contact problems, with appropriate boundary conditions. In discrete form this uses the information from neighbouring points to define the elastic deflection at each point in the domain. The other advantage of this method is that it yields additional solution information such as displacement, and derived fields such as stress, throughout the solid components, which is not possible using traditional half-space approach as discussed above. A view of the 3D domain Ω, showing the fluid boundary (Ω_f) and the bottom boundary (Ω_D), is given in the Figure 2.6. In [50] it is demonstrated that a geometry of size $60 \times 60 \times 60$ (non-dimensional) is sufficiently large to provide solutions for which the elastic deformation at the contact is no longer dependent on the domain size. Hence this is adopted throughout

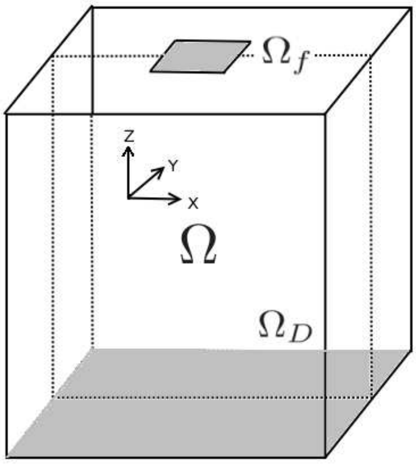


Figure 2.6: A view of the 3D elasticity domain Ω showing Ω_f (the fluid boundary) and Ω_D (the bottom boundary)

this thesis. Note that a cross-section of the 3D domain (the XZ-plane where $Y = 0$) defines the 2D elasticity domain for the line contact problems.

The linear elasticity equation reads [76]:

$$\frac{\partial}{\partial x_j}\left(c_{ijkl}\frac{\partial u_k}{\partial x_l}\right) = -F_i, \tag{2.30}$$

where F_i is the body force e.g. due to gravity or other external forces, and repeated suffices imply summation over the number of space dimensions. The fourth order tensor is defined by

$$c_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \tag{2.31}$$

and λ and μ (known as Lamé's coefficients) are material properties given by

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)} \, .$$

Here δ_{ij} is the Kronecker delta, whilst E is the Young's modulus and ν is the Poisson ratio of the solid component that is being deformed. In EHL the body force is assumed to be zero, with all of the non-negligible contributions to the elastic deformation coming from the high pressure at the contact, so equation (2.30) can be written as:

$$\frac{\partial}{\partial x_j}\left(c_{ijkl}\frac{\partial u_k}{\partial x_l}\right) = 0. \tag{2.32}$$

The resolution of equation (2.32) in a domain Ω is subject to the following boundary conditions [51]:

$$\begin{cases} u = 0 & \text{at the bottom boundary } \Omega_D; \\ \sigma_n = n_j c_{ijkl} \frac{\partial u_k}{\partial x_l} = -\delta_{id} p & \text{at the fluid boundary } \Omega_f; \\ \sigma_n = 0 & \text{elsewhere}, \end{cases} \tag{2.33}$$

where d represents the dimension of the elasticity domain.

2.6.1 Non-dimensionalisation

In order to get a suitable non-dimensional form of equation (2.32), let us define dimensionless variables as follows:

$$X_i = \frac{x_i}{a}, \qquad U_k = \frac{u_k R}{a^2} \qquad P = \frac{p}{p_h} \text{ and } \qquad C_{ijkl} = c_{ijkl} \times \frac{a}{R p_h}\ .$$

Hence equation (2.32) can be written as:

$$\frac{\partial}{a \partial X_j} \left(\frac{R p_h}{a} C_{ijkl} \frac{\partial \left(\frac{a^2 U_k}{R} \right)}{a \partial X_l} \right) = 0,$$

or

$$\frac{\partial}{\partial X_j} \left(C_{ijkl} \frac{\partial U_k}{\partial X_l} \right) = 0, \tag{2.34}$$

where the corresponding boundary conditions (2.33) become:

$$\begin{cases} U = 0 & \text{at the bottom boundary } \Omega_D; \\ \sigma_n = n_j C_{ijkl} \frac{\partial U_k}{\partial X_l} = -\delta_{id} P & \text{at the fluid boundary } \Omega_f; \\ \sigma_n = 0 & \text{elsewhere}. \end{cases} \tag{2.35}$$

2.6.2 Equivalent problem

In [51], it is demonstrated that the two contacting solid components with material properties (E_1, ν_1) and (E_2, ν_2) can be reduced to an equivalent solid component with equivalent material properties (E_{eq}, ν_{eq}) such that:

$$E_{eq} = \frac{E_1^2 E_2 (1 + \nu_2)^2 \ + \ E_2^2 E_1 (1 + \nu_1)^2}{\{E_1 (1 + \nu_2) \ + \ E_2 (1 + \nu_1)\}^2} \tag{2.36}$$

$$\nu_{eq} = \frac{E_1\nu_2(1+\nu_2)\ +\ E_2\nu_1(1+\nu_1)}{E_1(1+\nu_2)\ +\ E_2(1+\nu_1)}. \tag{2.37}$$

Moreover, the non-dimensional elasticity tensor for equivalent problem implies:

$$C_{ijkl} = c_{ijkl}(E_{eq}, \nu_{eq}) \times \frac{a}{Rp_h} = c_{ijkl}(E_{eq} \times \frac{a}{Rp_h}, \nu_{eq}) = c_{ijkl}(E_{new}, \nu_{eq}). \tag{2.38}$$

Under the same pressure distribution in Ω_f, the solution of the equivalent elasticity problem gives the total elastic deformation of both contacting solids [51]. If both contact surfaces have same material properties (E, ν) then the equivalent Young's modulus and Poisson's ratio for the equivalent problem reduces to:

$$E_{eq} = \frac{E}{2} \quad \text{and} \quad \nu_{eq} = \nu.$$

A simplified expression of E_{new} in equation (2.38) can be obtained by using expressions of the Hertzian half contact width a and the maximum Hertzian pressure p_h defined for the line contact problems:

$$E_{new} = E_{eq} \times \frac{a}{Rp_h} = \frac{E}{2} \times \frac{a.\pi.a}{2wR} = \frac{\pi E}{4wR} \times a^2 = \frac{\pi E}{4wR} \times \frac{8wR}{\pi E'} = \frac{2E}{E'},$$

where

$$E' = \frac{E}{1-\nu^2}.$$

Hence

$$\begin{aligned} E_{new} &= 2E \times \frac{(1-\nu^2)}{E} \\ &= 2(1-\nu^2). \end{aligned} \tag{2.39}$$

For the point contact case, a similar procedure can be applied to define E_{new} by using expressions of the Hertzian radius a and the maximum Hertzian pressure p_h defined for the point contact problems:

$$E_{new} = E_{eq} \times \frac{a}{Rp_h} = \frac{E}{2} \times \frac{2\pi a^3}{3FR} = \frac{\pi E}{3FR} \times a^3 = \frac{\pi E}{3FR} \times \frac{3FR}{2E'} = \frac{\pi E}{2E'},$$

using the definition of E' given above implies:

$$E_{new} = \frac{\pi}{2}(1-\nu^2). \tag{2.40}$$

By using E_{new}, the dimensionless total elastic deformation of the contacting solid components is obtained by solving equation (2.34) subject to the boundary conditions (2.35) in appropriate domains. The elastic deflection $D(X)$ in equation (2.17) and $D(X, Y)$ in equation (2.23) is related to the displacement field $\mathbf{U}$ through the following relation:

$$D = -U_z \mid_{\Omega_f} .$$

2.7 Summary

In this chapter, the line and the point contact EHL equations are introduced, followed by their non-dimensional forms. The equations (2.16), (2.17), (2.19), (2.26) & (2.27) define a complete non-dimensional isothermal steady state EHL line contact problem. On the other hand the equations (2.22), (2.23), (2.25), (2.26) & (2.27) define a complete non-dimensional isothermal steady state EHL circular point contact problem. Note that due to dependencies of density and viscosity relationships upon the pressure, the Reynolds equation is highly nonlinear. Moreover, two ways are described to calculate the elastic deformation: the first, so-called the half-space approximation of the linear elasticity equation, which is given in the relations (2.18) & (2.24) for the line and the point contact cases respectively, whilst the latter is concerned with the numerical solution of the Lamé's equation of linear elasticity in appropriate domains [49–51]. Throughout this thesis, the latter approach is considered for the elastic deformation solution and the issue of the high computational and memory costs of this approach to solve EHL problems are discussed. Indeed, the primary focus of this thesis is to develop fast, efficient and reliable numerical techniques for the solution of this problem. Since our concern is primarily with these numerical methods we focus exclusively on the equivalent reduced geometry problem in the remainder of this work.

Chapter 3

Numerical Methods

3.1 Introduction

In this chapter we discuss various numerical methods used throughout this thesis for the solution of discretized forms of steady-state EHL problems. In the first section, a brief introduction to the finite element method (FEM) is provided. Later, it is demonstrated that the discretized forms of EHL problems reduce to systems of nonlinear algebraic equations. The Newton method is described for obtaining the numerical solution of systems of nonlinear equations, whilst for the solution of linear systems arising at each Newton iteration both direct and iterative approaches are discussed. Later in this chapter multigrid methods [21, 100] are explained to give a general introduction of this class of solution method. Finally, a description of an open source software library [57] is provided since this is used as the framework for the implementation of different techniques discussed throughout this chapter.

3.2 Finite Element Method

Many physical phenomena in science and engineering, e.g. fluid dynamics, solid mechanics, electromagnetics, biomechanics, etc., can be represented in terms of partial differential equations. The finite element method (FEM) [18, 35, 40, 93, 109] is a numerical technique to find the approximate solution of such partial differential equations. The basic idea of the finite element method is to divide the domain (on which the equation is prescribed) into a number of finite elements, consisting of simple convex shapes whose vertices are known as nodes, to obtain an approximate solution. The division of the domain into finite elements is known as finite element mesh and the process is itself called mesh generation.

By way of motivation, the finite element method is explained for a simple elliptic partial differential equation in a domain $\Omega \subset \mathbb{R}^d$ ($d = 1, 2,$ or 3) with $\partial\Omega$ representing its boundary. The Poisson equation [35] involves in finding a solution $u \in \mathbb{R}^d$ such that:

$$-\nabla^2 u = f \qquad \text{in } \Omega, \tag{3.1}$$

subject to the boundary conditions:

$$\begin{aligned} u &= g_0 && \text{on } \Gamma_D \neq \phi \\ \nabla u.\vec{n} &= g_1 && \text{on } \Gamma_N, \end{aligned} \tag{3.2}$$

where $\nabla^2 = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$ is a d-dimensional Laplacian operator in cartesian coordinates, $f \subset \mathbb{R}^d$ is the source function, $\Gamma_D \cup \Gamma_N = \partial\Omega$, whilst $\vec{n}$ is a unit normal to the boundary $\partial\Omega$. Note that where the value of u is specified on the part of boundary Γ_D these are known as Dirichlet boundary conditions, while in the remaining part of the boundary $\Gamma_N = \partial\Omega \setminus \Gamma_D$ the conditions specified in terms of the normal derivative are known as the Neumann boundary conditions. The finite element approximation consists of replacing the strong form of the problem by a weak form [35, 93]. This is obtained by multiplying an appropriate test function $v \in H_0^1(\Omega)$ (see below) on both sides of equation (3.1) and integrating over the domain Ω. This yields:

$$-\int_{\Omega} v\nabla^2 u \mathrm{d}\Omega = \int_{\Omega} vf \mathrm{d}\Omega, \tag{3.3}$$

applying the Green's formula [93] on the left-hand side yields the required weak formulation:

$$\int_{\Omega} \nabla u.\nabla v \mathrm{d}\Omega - \int_{\partial\Omega} (\nabla u.\vec{n}) v \, \mathrm{d}s = \int_{\Omega} fv \mathrm{d}\Omega, \tag{3.4}$$

or

$$\int_{\Omega} \nabla u.\nabla v \mathrm{d}\Omega = \int_{\Omega} fv \mathrm{d}\Omega + \int_{\partial\Omega} (\nabla u.\vec{n}) v \, \mathrm{d}s. \tag{3.5}$$

Before proceeding to next step, let us consider the above more rigorously by first defining the space of square integrable functions $L^2(\Omega)$ [18, 35], given by

$$L^2(\Omega) := \left\{ u : \Omega \to \mathbb{R} \mid \int_\Omega u^2 \mathrm{d}\Omega < \infty \right\},$$

where the norm associated with this space (known as the L_2 norm) is given by

$$\|u\|_2 = \|u\|_{L^2(\Omega)} = \left(\int_\Omega u^2 \mathrm{d}\Omega \right)^{\frac{1}{2}} .$$

Now the integrals in the weak form will be well defined if the dot product of the gradients of the functions u and v, and the functions f, v and g_1 are square integrable. Such functions are the members of a well known space called the Sobolev space $H^1(\Omega)$ [18, 35] which is defined as:

$$H^1(\Omega) := \left\{ u : \Omega \to \mathbb{R} \mid u, \frac{\partial u}{\partial x_i} \in L^2(\Omega) \text{ for } i = 1, .., d \right\} .$$

The space does not take into account the Dirichlet boundary conditions thus the solution space is defined as

$$H^1_E(\Omega) := \left\{ u \in H^1(\Omega) \mid u = g_0 \text{ on } \Gamma_D \right\}$$

where, as noted above, the test space should be

$$H^1_0(\Omega) := \left\{ v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D \right\} .$$

Now, equation (3.5) can be written more precisely as:

$$\int_\Omega \nabla u . \nabla v \mathrm{d}\Omega = \int_\Omega f v \mathrm{d}\Omega + \int_{\Gamma_N} g_1 v \, \mathrm{d}s. \tag{3.6}$$

The finite element method consists of approximating u by replacing the weak problem by a finite-dimensional problem. This is achieved by taking finite-dimensional subspaces: $S^h_E \subset H^1_E$ and $S^h_0 \subset H^1_0$ [35]. For this purpose, the domain Ω is divided into a set of m non-overlapping elements interconnected at $n+n_D$ discrete nodes (where n is the number of nodes in $\Omega \setminus \Gamma_D$ and n_D is the number of nodes on Γ_D). This then defines the finite-dimensional subspace: $S^h_0 = \mathrm{span}\{N_1, N_2, ..., N_n\}$ of piecewise functions which are continuous across the whole domain Ω and zero on the Dirichlet boundary. In the simplest case (which is the focus of this thesis) each of these elements is a simplex (triangle in 2D, tetrahedron in 3D) and the functions in the basis set are uniquely associated with each node P_j in $\Omega \setminus \Gamma_D$ for $j = 1, 2, .., n$, and satisfy the following properties [93]:

$$N_j(P_i) = \begin{cases} 1 & \text{if } P_i = P_j \\ 0 & \text{if } P_i \neq P_j\ . \end{cases} \tag{3.7}$$

For the finite-dimensional solution subspace S_E^h the basis set is extended by adding another n_D basis functions which correspond to the nodes on the Dirichlet boundary Γ_D in order to take into account the Dirichlet boundary conditions [35]. Since the set of the basis functions of the test subspace is subset to that of the basis set of the solution subspace so the kind of approximation is generally referred to as the Galerkin approximation [18, 35, 40, 109]. The Galerkin finite element approximation: $u^h \in S_E^h$ is of the form:

$$u^h = \sum_{i=1}^{n+n_D} u_i\, N_i$$

or

$$u^h = \sum_{i=1}^{n} u_i\, N_i + \sum_{i=n+1}^{n+n_D} u_i\, N_i, \tag{3.8}$$

where the values of $u_1, u_2, ..., u_n$ are unknowns while $u_{n+1}, u_{n+2}, ..., u_{n+n_D}$ are given by the Dirichlet boundary conditions. So replacing u in equation (3.6) with the approximation given in (3.8) and v with N_j for $j = 1, 2, ..., n$, we have a system of n equations in n-unknowns which is given by

$$\sum_{i=1}^{n} u_i \int_\Omega \nabla N_i.\nabla N_j \mathrm{d}\Omega = \int_\Omega f N_j \mathrm{d}\Omega + \int_{\Gamma_N} g_1 N_j\, \mathrm{d}s - \sum_{i=n+1}^{n+n_D} u_i \int_\Omega \nabla N_i.\nabla N_j \mathrm{d}\Omega\ ,$$

for $j = 1, 2, ..., n$. In particular when $u = g_0 = 0$ on Γ_D, then S_E^h is same as S_0^h in the Galerkin approximation, and also the above system reduces to

$$\sum_{i=1}^{n} u_i \int_\Omega \nabla N_i.\nabla N_j \mathrm{d}\Omega = \int_\Omega f N_j \mathrm{d}\Omega + \int_{\Gamma_N} g_1 N_j\, \mathrm{d}s\ . \tag{3.9}$$

The above system can typically be written in matrix notation as

$$K\mathbf{u} = \mathbf{f}, \tag{3.10}$$

with $\mathbf{u} = [\mathbf{u}_j]$ and

$$K = [K_{ji}], \qquad K_{ji} = \int_\Omega \nabla N_j.\nabla N_i \mathrm{d}\Omega$$

and

$$\mathbf{f} = [\mathbf{f}_j] = \int_\Omega f N_j \mathrm{d}\Omega + \int_{\Gamma_N} g_1 N_j \, \mathrm{d}s.$$

(Note that if $g_0 \neq 0$ the only change is some additional contributions to certain components of the right-hand side **f**.)
The matrix K is referred to as the global stiffness matrix. Note that the entry K_{ji} of the global stiffness matrix K will always be zero unless the nodes associated with the basis functions N_j and N_i belong to the same element of the mesh. This leads to the observation that the global stiffness matrix will be sparse, and the j-th row of the global stiffness matrix will only have the nonzero entries in its i-th column if the nodes j and i belong to the same element. In other words, one need to only think of a patch of elements sharing a common node j. Therefore, in practice, it is useful to calculate the element stiffness matrices for each element separately and then summing up their contributions to the global stiffness matrix by

$$K_{ji} = \int_\Omega \nabla N_i . \nabla N_j \mathrm{d}\Omega = \sum_{e=1}^{m} \int_{\Omega_e} \nabla N_i . \nabla N_j \mathrm{d}\Omega_e$$

This process is known as an assembly process [93]. A similar procedure is to be carried out for the right hand side vector **f** i.e.

$$\mathbf{f} = \sum_{e=1}^{m} \mathcal{I}_e \mathbf{f_e} \qquad \text{where} \qquad \mathcal{I}_e \in \mathbb{R}^{n \times (d+1)}$$

where

$$\mathbf{f_e} = \int_{\Omega_e} f N_j \mathrm{d}\Omega_e + \int_{\Gamma_N \cap \partial\Omega_e} g_1 N_j \, \mathrm{d}s.$$

Finally, the finite element meshing of $\Omega \subset \mathbb{R}^d$ may involve: line elements ($\mathbb{R}$), triangles/rectangles ($\mathbb{R}^2$) or tetrahedra/bricks ($\mathbb{R}^3$) [35, 109] etc. In this work, triangular ($\mathbb{R}^2$) and tetrahedral elements ($\mathbb{R}^3$) are used. The basis functions discussed above can be approximated to an arbitrary accuracy using polynomials which generally depend upon the number of nodes used in an element. For convenience, we only consider the two-dimensional case in detail here, i.e. $\Omega \subset \mathbb{R}^2$. A triangular FE mesh of Ω is shown in the Figure (3.1) with an element in zoom-in interconnected at three nodes numbered locally from 1 to 3 in anti-clockwise direction. The polynomial basis functions defined on the three node triangular element are linear and are given by [109]:

$$N_I^e = \frac{a_I + b_I x + c_I y}{2A} \qquad \text{for} \qquad I = 1, 2, 3,$$

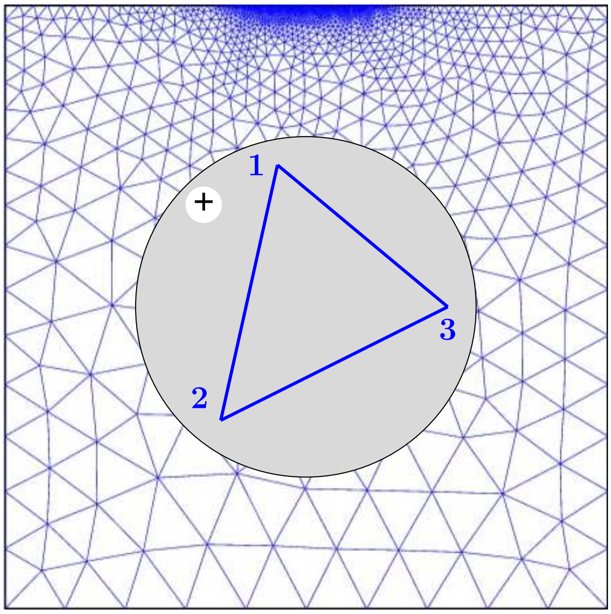


Figure 3.1: A triangulation

with

$$\begin{aligned} a_1 &= x_2y_3 - x_3y_2, & b_1 &= y_2 - y_3, & c_1 &= x_3 - x_2 \\ a_2 &= x_3y_1 - x_1y_3, & b_2 &= y_3 - y_1, & c_2 &= x_1 - x_3 \\ a_3 &= x_1y_2 - x_2y_1, & b_3 &= y_1 - y_2, & c_3 &= x_2 - x_1, \end{aligned}$$

while A^e is area of element:

$$A^e = \frac{a_1 + a_2 + a_3}{2}.$$

For further details about different types of elements and the approximation: the reader is referred to standard finite element texts, for example, see [109].

Finally, note that the finite element discretization of nonlinear partial differential equations leads to systems of nonlinear algebraic equations (this is discussed in detail in Chapter 4).

3.3 Newton Method

Newton's method [68] is considered as a powerful numerical tool for solving systems of nonlinear algebraic equations. These systems arise in many applications, including the discretization of nonlinear partial differential equations. Consider a system of n nonlinear algebraic equations:

$$\mathbf{F}(\mathbf{u}) = \mathbf{0}, \tag{3.11}$$

here the residual $\mathbf{F} : R^n \rightarrow R^n$ and $\mathbf{u}$ is a vector of n-unknowns, expressed as:

$$\mathbf{u} = [u_1, u_2, ..., u_n]^T.$$

Alternatively, equation (3.11) can be written as:

$$F_i(\mathbf{u}) = 0 \qquad\qquad i = 1, 2, ..., n. \tag{3.12}$$

In the neighbourhood of $\mathbf{u}$, Taylor series can be used to expand each function F_i as:

$$F_i(\mathbf{u} + \delta\mathbf{u}) = F_i(\mathbf{u}) + \sum_{j=1}^{n} \frac{\partial F_i}{\partial u_j} \delta u_j + h.o.t. \qquad i = 1, 2, ..., n. \tag{3.13}$$

In vector notation this is:

$$\mathbf{F}(\mathbf{u} + \delta\mathbf{u}) = \mathbf{F}(\mathbf{u}) + \mathbf{F}'(\mathbf{u})\delta\mathbf{u} + h.o.t. \tag{3.14}$$

Ignoring the higher order terms ($h.o.t.$), equation (3.14) can be written as:

$$\mathbf{F}(\mathbf{u}) + \mathbf{F}'(\mathbf{u})\delta\mathbf{u} \approx \mathbf{0},$$

in the case where we wish to solve $\mathbf{F}(\mathbf{u} + \delta\mathbf{u}) = \mathbf{0}$. This re-arranges to

$$\mathbf{F}'(\mathbf{u})\delta\mathbf{u} \approx -\mathbf{F}(\mathbf{u}), \tag{3.15}$$

where $\mathbf{F}'(\mathbf{u})$ is the system Jacobian:

$$\mathbf{F}'(\mathbf{u}) = \begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} & .... & \frac{\partial F_1}{\partial u_n} \\ \frac{\partial F_2}{\partial u_1} & \frac{\partial F_2}{\partial u_2} & .... & \frac{\partial F_2}{\partial u_n} \\ : & : & : & : \\ \frac{\partial F_n}{\partial u_1} & \frac{\partial F_n}{\partial u_2} & .... & \frac{\partial F_n}{\partial u_n} \end{bmatrix}.$$

Solving equation (3.15) at an initial guess $\mathbf{u}^{old}$, for the correction $\delta\mathbf{u}$, yields an update:

$$\mathbf{u}^{new} = \mathbf{u}^{old} + \delta\mathbf{u}.$$

Sometimes this full Newton approach fails to converge (if $\mathbf{u}^{old}$ is not sufficiently close to the true solution) and so damping of the Newton update is often used in order to improve convergence of the Newton iteration:

$$\mathbf{u}^{new} = \mathbf{u}^{old} + \lambda \delta \mathbf{u},$$

where $\lambda \in (0, 1]$ is often known as a damping factor or step length. The whole Newton process can be summarized in the following steps:

1. Set $\mathbf{u}^0$ = an initial guess.
2. For $k = 0, 1, 2, ...$ until convergence, repeat:
   (a) Solve
   $$\mathbf{F}'(\mathbf{u}^k)\, \delta \mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k)$$
   for the Newton step $\delta \mathbf{u}^k$.
   (b) Set $\mathbf{u}^{k+1} = \mathbf{u}^k + \lambda \delta \mathbf{u}^k$, where $0 < \lambda \leq 1$.
   (c) Test for convergence.

A significant advantage of Newton's method is that it is generally expected to achieve quadratic convergence provided the initial guess is good enough for it to converge. However, if the initial guess is not sufficiently good then the Newton iteration may diverge. The computation of the Newton step is most expensive part of the Newton iteration since a linear system must be solved. This may be found by means of a direct method or an iterative method.

Generally, it is not necessary to solve the linear system exactly at each Newton iteration, i.e. it is only necessary to solve the linear system to a sufficient precision in order to determine a good enough Newton update to achieve convergence of the nonlinear system. The **inexact Newton method** makes use of an iterative method for the solution of the linear system at each Newton iteration so that it may be solved approximately. One way of implementing this is to choose the stopping criterion for the Newton step as:

$$\|\mathbf{F}'(\mathbf{u}^k)\, \delta \mathbf{u}^k + \mathbf{F}(\mathbf{u}^k)\| \leq \eta_k \|\mathbf{F}(\mathbf{u}^k)\| \,, \tag{3.16}$$

where the variable η_k is called a forcing term. The best choice of this parameter is highly empirical, however different choices are suggested in, for example, [34]. A large value of this parameter may leads to a poor determination of the Newton step which can cause divergence. On the other hand a relatively very small value of this parameter, although leading to fewer Newton iterations, leads to more computational work overall. It should be noted that in this approach the accuracy of the inner solves increases whenever the solution comes closer to solution of the nonlinear system.

3.4 Sparse Direct Method

Direct methods for solving linear systems may be defined as the methods that seek the exact solution of a linear system in a finite number of steps (in the absence of rounding errors). The direct methods generally involve algorithms that reduce the system matrix to some special and easily solvable form e.g. a triangular form or equivalently to the product of lower triangular L and upper triangular U factors. This is known as LU decomposition [26]. The discretization techniques such as the finite element discretization of partial differential equations generally leads to large and very sparse system matrices. Efficient direct algorithms for sparse matrices are much more complicated than for the dense matrices [32]. The main complication is due to the need for efficiently handling the fill-in in the factors L and U. Note that fill-in are the nonzero entries in the factors L and U that are not present in the original matrix. The fill-in significantly affects the efficiency of a direct algorithm both in terms of computational time and memory requirements. A typical direct algorithm [32] assumes the solution in the following four phases:

1. A pre-ordering phase that applies a suitable algorithm to re-order the rows and columns of system matrix in order to reduce fill-in. This is normally independent of the numeric values of the system matrix.

2. Symbolic analysis which determines the nonzero structures of the factors L and U and creates suitable data structures for these factors. This helps to predict the memory requirement for the numerical factorization. However, this is only an estimate, the actual requirement depends upon pivoting chosen (for stability) during the numerical factorization.

3. Numerical factorization which computes the factors L and U of the original matrix.

4. A solve phase that computes the solution of the system by performing two triangular solves using the factors computed in step 3.

Sometime an additional phase, iterative refinement, is used after the solution phase to improve the accuracy of the solution. This only requires repetition of step 4 above but using the residual on the right-hand side. Numerical factorization is generally the most time consuming phase while the solve phase is about an order of magnitude faster [32]. There are a number of algorithms associated with each phase which have their own impact on the performance of a direct algorithm for some class of problems. A number of efficient sparse direct algorithms [26, 27] have been developed so far and further work is still going on in this field. Amongst them, UMFPACK [27–30] is a public domain software which is used throughout this thesis. UMFPACK is based on the Unsymmetric-pattern MultiFrontal method. This combines the step 1 & 2 and thus assumes the solution only in three steps. UMFPACK factorizes PAQ, $PRAQ$, or $PR^{-1}AQ$ into the product LU, where P and Q are permutation matrices which reduces the fill-in. Importantly, the other role of P is to maintain the numerical accuracy of the solution (hence this depends upon numerical values and not just the sparsity pattern). Finally, R is a diagonal matrix of row scaling factors. For more details, one is referred to [27].

The use of a direct method to determine a Newton step may be a good choice if the Jacobian can be computed and stored efficiently and the cost of factorization of the Jacobian is not excessive or the Jacobian is dense (rendering iterative methods ineffective). Another possible reason of the choice of a direct method may be the failure of convergence of iterative methods for a specific problem. Despite the fact of being more expensive in terms of Jacobian factorization and storage requirements, sparse direct methods are still used in real applications due to their robustness.

In some applications the discretization of 1D problems often leads to tridiagonal systems which can be efficiently solved using **Thomas' algorithm** [97]. The said algorithm is a simplified version of Gaussian elimination method which requires only $\mathcal{O}(n)$ operations to find the exact solution [97]. This method employs in two steps: in the first step all the nonzero entries in the diagonal below the main diagonal are eliminated while the second step the backward substitution is used to obtain the solution. It may be extended for any banded matrix for which the bandwidth is independent of n: exploiting the fact that no fill-in occurs beyond the band structure.

3.5 Newton Krylov Methods

Recall from the previous section that an inexact Newton strategy uses an iterative linear approach for the inner solve to approximate a Newton step. As the name suggests a Newton Krylov method uses a Krylov subspace based iterative technique for its inner linear solve. For a solution of a linear system $Ax = b$, a Krylov subspace linear iterative solver seeks the approximation of the form:

$$x_k = x_0 + \tilde{x},$$

where x_0 is an initial iterate and $\tilde{x} \in \mathcal{K}_k$, a kth Krylov subspace spanned by a set of k vectors $\{r_0, , Ar_0, ..., A^{k-1}r_0\}$ with $r_0 = b - Ax_0$ the initial residual. Thus a Krylov subspace method involves matrix-vector products and builds an approximation from an appropriate Krylov subspace. In the following sections a few such methods are discussed which have been frequently used throughout this work.

3.5.1 Conjugate Gradient Method

The conjugate gradient method (CG) is the most well known method from the general family of Krylov subspace methods. It was originally developed as a direct method for symmetric positive definite (SPD) systems in 1952 by Hestenes and Stiefel [56] but was not widely used due to being considered computationally expensive. Later in 1971, Reid [89] revived it as an iterative method for large sparse SPD systems. For a symmetric positive definite matrix A, solving $Ax = b$ is equivalent to minimizing the following quadratic function:

$$f(x) = \frac{1}{2}x^TAx - x^Tb. \tag{3.17}$$

This minimum occurs when the gradient is zero:

$$\nabla f(x) = \frac{1}{2}A^Tx + \frac{1}{2}Ax - b = 0,$$

and since $A^T = A$,

$$Ax - b = 0.$$

The CG is basically a line-search method which seeks to minimize a one-dimensional problem by finding the value of α_k that minimizes $f(x_{k+1})$ where:

$$x_{k+1} = x_k + \alpha_k p_k, \tag{3.18}$$

and the vector p_k denotes the search direction from the existing iterate x_k. Using relation (3.18), the following recurrence relation holds for the residual vectors:

$$r_{k+1} = b - Ax_{k+1} = r_k - \alpha_k Ap_k. \tag{3.19}$$

Various strategies are possible to determine the search directions. Recall from linear algebra that a gradient of a function defines a direction in which the function increases most rapidly, and hence the negative gradient provides the direction of steepest descent. Such a choice of a direction yields the so-called steepest descent method [44, 93]. Note that the negative gradient of the function (3.17) at a given approximation x_k defines the residual $r_k = b - Ax_k$, therefore the iteration (3.18) becomes:

$$x_{k+1} = x_k + \alpha_k r_k.$$

The drawback of this method is its slow convergence for the ill-conditioned systems. Another approach is to generate a set of search directions $\{p_0, p_1, ..., p_{n-1}\}$ which are conjugate to each other with respect to A (i.e. A-orthogonal). The A-orthogonality condition reads:

$$p_i^T A p_j = 0 \quad \forall \quad i \neq j.$$

An important property of such a set is that all n vectors are linearly independent with respect to the inner product generated by A. Methods using such a set of directions are called conjugate direction methods [24]. The Gram-Schmidt [93] conjugation generates a set of conjugate directions from a set of n linearly independent vectors $\{u_0, u_1, ..., u_{n-1}\}$, however the disadvantage of this method is its computational complexity as it requires the storage of all previous search directions. This computational complexity is reduced significantly by conjugate gradient method [24,35,44,93] which uses a simple recurrence relation to define the search directions. As the name suggests, this is basically a conjugate direction method which involves producing search directions by conjugation of residuals. The recurrence relation used to define such search directions is basically a linear combination of the residual r_{k+1} and the previous search direction p_k i.e.

$$p_{k+1} = r_{k+1} + \beta_k p_k, \tag{3.20}$$

where the parameter β_k is chosen such that the new search direction p_{k+1} is A-orthogonal to previous search direction p_k and is given by:

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}. \tag{3.21}$$

Putting all the steps together constitutes the Algorithm 1.

Note that the CG method only requires the storage of the last two residual vectors, one last search direction and only one matrix vector multiplication per iteration. Hence the computational cost per iteration is $\mathcal{O}(m)$ where m is the number of non-zero entries

Algorithm 1 Conjugate Gradient Method [35]

```
compute $p_0 = r_0 = b - Ax_0$
**for** $k = 0, 1, 2, ....$until convergence (see (3.16)), **do**
    $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$
    (test for convergence)
    $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
    $p_{k+1} = r_{k+1} + \beta_k p_k$
**end for**
```

of A. In short, the conjugate gradient method is considered as a powerful numerical tool for solving problems which involve large and sparse symmetric positive definite system matrices.

3.5.2 Generalized Minimal Residual Method

The generalized minimal residual method (GMRES) [94] was designed by Saad and Schultz in 1986, to solve linear systems $Ax = b$ with unsymmetric coefficient matrices A. When A is unsymmetric, then minimizing the function (3.17) is no longer equivalent to solving $Ax = b$. In the GMRES method [94] the minimizing functional in the Krylov subspace $\mathcal{K}_k(A, r_0)$ is chosen as the 2-norm of the residual r_k. Starting with x_0 as an initial guess and defining $r_0 = b - Ax_0$ with $v_1 = r_0/\|r_0\|_2$, the GMRES method uses the Arnoldi method [93, 94] implemented through modified Gram-Schmidt orthogonalization which sequentially generates orthonormal basis vectors $\{v_1, v_2, ..., v_k\}$, such that:

$$\mathrm{span}\{v_1, v_2, ..., v_k\} = \mathrm{span}\{v_1, Av_1, ..., A^{k-1}v_1\} = \mathcal{K}_k(A, v_1).$$

Let V_k denote a matrix with v_j in its j-th column and H_k denote a $k \times k$ upper Hessenberg matrix formed by the numbers h_{ij} computed at the same time by Arnoldi method [93], then the reduction of A to Hessenberg form is given by [93, 94]:

$$H_k = V_k^T A V_k.$$

The relation between V_k and V_{k+1} is given by:

$$AV_k = V_{k+1}\bar{H}_k, \tag{3.22}$$

where $\bar{H}_k$ is $(k+1) \times k$ upper Hessenberg matrix:

$$\bar{H}_k = \begin{bmatrix} H_k \\ \mathbf{h}_{k+1} \end{bmatrix} \quad ; \; \mathbf{h}_{k+1} = (0, 0, ..., h_{(k+1)k}).$$

The k-th GMRES approximation seeks to approximate the solution of the form:

$$x_k = x_0 + \tilde{x},$$

where $\tilde{x} \in \mathcal{K}(A, r_0)$ and this is determined by using the first k-orthonormal vectors generated by the algorithm, i.e:

$$\tilde{x} = V_k y,$$

therefore

$$x_k = x_0 + V_k y, \tag{3.23}$$

where y is a k-vector which is chosen to minimize the L_2-norm of the corresponding residual ($\|r_k\|_2$) [93, 94], where

$$\begin{aligned} \|r_k\|_2 &= \|b - Ax_k\|_2 \\ &= \|b - A(x_0 + V_k y)\|_2 \\ &= \|r_0 - AV_k y\|_2, \end{aligned}$$

using the relation (3.22), we have

$$\begin{aligned} \|r_k\|_2 &= \|\beta v_1 - V_{k+1}\bar{H}_k y\|_2 \qquad ; \beta = \|r_0\|_2 \\ &= \|V_{k+1}\left(\beta e_1 - \bar{H}_k y\right)\|_2 \;, \end{aligned}$$

since columns of V_{k+1} are orthonormal so the functional $f(y)$ to be minimized is equivalent to, see [93, 94]:

$$f(y) = \|r_k\|_2 = \|\beta e_1 - \bar{H}_k y\|_2 \;. \tag{3.24}$$

Here $e_1 = \{1, 0, 0, ..., 0)$ is a unit vector of length $k+1$. Finally the GMRES algorithm [94] requires a solution of $(k+1) \times k$ least square problem to obtain the optimal value of y. A summary of all the above steps is given in Algorithm 2.

Assuming the use of exact arithmetic, GMRES converges in at most n steps [94]. Note that at each GMRES iteration, an orthogonalization of a vector v_{k+1} to $\mathcal{K}_k(A, v_1)$ is required against all the previous constructed vectors $\{v_i\}_{i=1}^{k}$. This grows the computational cost at each iteration like $\mathcal{O}(kn)$. In other words an increase in the number of GMRES

Algorithm 2 GMRES [93]

```
1:  Compute $r_0 = b - Ax_0,\ \beta := \|r_0\|_2$, and $v_1 := \frac{r_0}{\beta}$
2:  for $j = 1, 2, ..., k$, ...until satisfied, do
3:      compute $w_j := Av_j$
4:      for $i = 1, ...., j$ do
5:          $h_{ij} := (w_j, v_i)$
6:          $w_j := w_j - h_{ij}v_i$
7:      end for
8:      $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$, set $k := j$ and exit loop (go to step $11$)
9:      $v_{j+1} = w_j/h_{j+1,j}$
10: end for
11: define the $(k+1) \times k$ Hessenberg matrix $\bar{H}_k = \{h_{ij}\}_{1 \leq i \leq k+1, 1 \leq j \leq k}$
12: compute $y$ the minimizer of $\|\beta e_1 - \bar{H}_k y\|_2$ and $x_k = x_0 + V_k y$.
```

steps increases both the memory usage and the computational cost, therefore, it becomes impractical whenever k is significantly large. The issues with large computational costs of this method is often avoided by restarting the iteration after a fixed number of steps, and this is known as restarted GMRES or GMRES(m), see [93, 94]. In the GMRES(m) version, if the convergence is not achieved for $k \leq m$ where m is specified as some upper bound of dimension of Krylov-subspace used, then GMRES is restarted with a new initial guess x_0 set to x_m. This process is repeated until convergence is achieved. Note however that a small value of m may lead to slow convergence of GMRES(m) or even entire failure of convergence, however the optimum choice tends to vary from one problem to another.

3.5.3 Other Iterative Techniques

The GMRES [94] method is very effective in solving general non-symmetric systems though at the cost of large storage requirements. The biconjugate gradient (BCG) method [93] is another approach for non-symmetric systems which produces two mutually orthogonal sequence of residuals, however providing no minimization of residual any longer unlike the GMRES. In other words, taking $v_1 = r_0/\|r_0\|_2$, this method consists of a projection process onto

$$\mathcal{K}_k(A, v_1) = \text{span}\{v_1, Av_1, ..., A^{k-1}v_1\},$$

orthogonal to

$$\mathcal{L}_k(A^T, v_1) = \text{span}\{v_1, A^T v_1, ..., (A^T)^{k-1}v_1\}.$$

In 1980s, a variant of BCG so-called conjugate gradient squared algorithm (CGS) [101] was proposed in order to avoid usage of A^T, and to provide faster convergence rate with almost the same computational cost. However this method suffers with irregular behaviour in convergence. In order to overcome these negative effects van der Vorst [101] proposed another variant of BCG so-called biconjugate gradient stabilized method (Bi-CGSTAB).

3.6 Preconditioning

The previous section is concerned with different iterative approaches for solving the linear system $Ax = b$ resulting from linearization of some nonlinear system. Iterative methods for solving sparse linear systems are generally superior to direct approaches in terms of memory. However, with increasing problem sizes, not only does the amount of computational work to carry out a single iteration increase, but also the number of iterations required to achieve convergence typically grows [35]. Thus, when this growth in iterations is too great, iterative methods can suffer from a lack of robustness compared to direct methods. This lack of performance of an iterative method is generally due to an unfavourable distribution of the eigenvalues of the system matrix [35], causing the system to become more ill-conditioned as the system gets larger. Frequently, the performance of an iterative method can be improved significantly by using preconditioning [35, 44, 93]. The term preconditioning is regarded as a way of transforming the original linear system into an equivalent system, which has the same solution as the original one, but also has a more favourable spectrum [35] (i.e. the eigenvalues of the transformed system are tightly clustered within a small region compared to the original system). Additionally, a good preconditioner M is a matrix for which it is inexpensive to calculate the effect of its inverse over an arbitrary vector ν. A preconditioner M can be applied to a linear system in three different ways [93].

- Left preconditioning leads to preconditioned system:

$$M^{-1}Ax = M^{-1}b \tag{3.25}$$

- Right preconditioner leads to preconditioned system:

$$AM^{-1}u = b, \quad x = M^{-1}u \tag{3.26}$$

  which involves substitution of a new variable u for x.

- Split preconditioning can also be applied if M is available in factored form i.e $M = M_1M_2$, this defines the following preconditioned system:

$$M_1^{-1}AM_2^{-1}u = M_1^{-1}b, \quad x = M_2^{-1}u\ . \tag{3.27}$$

Recall from the previous section that the conjugate gradient method may be applied to systems with symmetric positive definite matrices. It should be noted that the preconditioned matrices $M^{-1}A$ and AM^{-1} are no longer necessarily symmetric positive definite [93] even if M is. Thus strategies preserving symmetry need to be considered. One way to preserve symmetry is the use of split preconditioning. Given a symmetric positive-definite matrix M, the availability $M = LL^T$ leads to preconditioned system:

$$L^{-1}AL^{-T}u = L^{-1}b, \quad x = L^{-T}u,$$

where the preconditioned system matrix $L^{-1}AL^{-T}$ is symmetric and positive-definite, therefore the conjugate gradient method is applicable to this system. Applying the conjugate gradient method to this system in a straightforward manner produces an algorithm which, at first appearance, requires L to be computed. However a few careful substitutions lead to preconditioned conjugate gradient [35, 44, 93], see Algorithm 3, which only requires one to compute the affect of M^{-1} over an arbitrary vector.

Algorithm 3 Preconditioned Conjugate Gradient Method [35]

```
compute $r_0 = b - Ax_0$, solve $Mz_0 = r_0$, set $p_0 = z_0$
for $k = 0, 1, 2, ....$until convergence, do
    $\alpha_k = \frac{z_k^T r_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$
    (test for convergence)
    solve $Mz_{k+1} = r_{k+1}$
    $\beta_k = \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$
    $p_{k+1} = z_{k+1} + \beta_k p_k$
end for
```

The GMRES algorithm [94] is developed for general systems thus the preconditioned GMRES algorithm can be obtained for any type of preconditioning. Recall from the previous section that the GMRES algorithm generates the minimization of the L_2-norm of the residual vector at each step. It should be noted that the left preconditioner modifies the residual vector while this is not the case in right preconditioning. In other words the GMRES algorithm minimizes the original residual in case of right preconditioning. In this section the preconditioned GMRES [93], see Algorithm 4, is only provided for right preconditioning which is used throughout this thesis.

Algorithm 4 Right Preconditioned GMRES [93]

```
1:  Compute $r_0 = b - Ax_0,\ \beta := \|r_0\|_2$, and $v_1 := \frac{r_0}{\beta}$
2:  for $j = 1, 2, ..., k, ...$until satisfied, do
3:      compute $w_j := AM^{-1}v_j$
4:      for $i = 1, ...., j$ do
5:          $h_{ij} := (w_j, v_i)$
6:          $w_j := w_j - h_{ij}v_i$
7:      end for
8:      $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$, set $k := j$ and exit loop (go to step $11$)
9:      $v_{j+1} = w_j/h_{j+1,j}$
10: end for
11: define the $(k+1) \times k$ Hessenberg matrix $\bar{H}_k = \{h_{ij}\}_{1\leq i\leq k+1, 1\leq j\leq k}$
12: compute $y$ the minimizer of $\|\beta e_1 - \bar{H}_k y\|_2$ and $x_k = x_0 + M^{-1}V_k y$.
```

As stated above, a good preconditioner M should be such that the resultant preconditioned (transformed) system has a tight clustered set of eigenvalues within a small interval (or a small number of small intervals), and additionally the effect of its inverse over an arbitrary vector is inexpensive to compute. A number of preconditioning techniques have been devised so far [13,25,35,93], among them [13] is a survey paper providing a review of different preconditioning techniques have been developed. Some common examples of preconditioning techniques include: diagonal preconditioners, incomplete LU factorization (ILU), sparse approximate inverse (SPAI) and multigrid preconditioners. The diagonal or Jacobi preconditioner consists of choosing the preconditioner M as the diagonal of the system matrix A. This preconditioner has only the effect of diagonal scaling of A. Moreover, if the system matrix is partitioned into blocks as a result of partitioning of unknown variables, then choosing only the diagonal blocks yields a preconditioner the so-called block diagonal preconditioner.

The incomplete LU factorization consists of computing approximate LU factors of the system matrix A such that the residual matrix $R = A - LU$ satisfies some specified constraints, such as having zeros in prescribed locations [93]. In other words, some constraints are applied to control the level of fill-in (see Section 3.4) in LU factors. In the simplest case where no fill-in is allowed in the factorization process is called no fill-in ILU or ILU(0). The accuracy of ILU(0) can be improved by allowing some fill-in during the factorization process. This involves by introducing a function called “fill-in level” to control the level of fill-in in the LU factors [13].

Sparse approximate inverse (SPAI) preconditioners involves computing the sparse approximation of A^{-1} explicitly and using this as a preconditioner for an iterative method [13, 25]. One simple way to compute a sparse approximate inverse M is by minimizing $\|I - AM\|$ in the Frobenius norm, subject to some sparsity constraints [13, 25, 93]. Thus computation of M involves solving n independent least square problems. Comprehensive details of this approach can be found in [25, 93].

Finally, we describe in details the multigrid preconditioning in the following section which includes a detailed introduction to multigrid methods and different variants of this approach.

3.7 Multigrid

Multigrid methods [21, 100] began to be developed in $1960s$ but efficient applications first arose with the work of Brandt [19]. A multigrid approach for EHL problems was first used by Lubrecht [73]. Later on Venner [102] and Goodyer [45] contributed in enhancing the efficiency of the application of multigrid methods to the Reynolds equation in EHL problems. Unlike these previous applications of multigrid to EHL problems, in this work multigrid methods are used as a preconditioner for the elasticity equation.

By way of motivation for this a general overview of multigrid methods is provided. For the sake of simplicity, let us assume that Ω_h represents a uniform grid of element size h, and $A_h x_h = b_h$ is a corresponding discrete system that needs to be solved on Ω_h. We further assume that $\tilde{x}_h$ is an approximation to the exact solution x_h obtained, for example, by applying few steps of a relaxation scheme (e.g. Jacobi, Gauss-Seidel, etc.). Then the error e_h can be defined as:

$$e_h = x_h - \tilde{x}_h,$$

and r_h is the residual:

$$r_h = b_h - A_h \tilde{x}_h.$$

The error and the residual can be related to each other through the error equation as follows:

$$\begin{aligned} A_h(\tilde{x}_h + e_h) &= b_h \\ A_h e_h &= b_h - A_h \tilde{x}_h \\ A_h e_h &= r_h. \end{aligned} \tag{3.28}$$

If the error equation is solved the solution can be corrected by:

$$x_h = \tilde{x}_h + e_h. \tag{3.29}$$

Note that the relaxation on the error equation (3.28) is equivalent to that on the original system [21]. So nothing can be obtained through this process that could not have been achieved by solving the original equation. However, if the error e_h in (3.29) can be approximated in some efficient way then an improved solution can be obtained through the correction:

$$\tilde{x}_h \leftarrow \tilde{x}_h + e_h. \tag{3.30}$$

Note that many relaxation schemes e.g. Jacobi, Gauss-Seidel, etc. have a smoothing effect on the error of approximation that means they eliminate the high frequency error components efficiently but damp slowly the low frequency error components [100]. Because of this property such relaxation methods are also known as smoothing methods or smoothers. On the other hand the smooth errors are damped more quickly on a coarser grid (for instance a grid with mesh size $2h$) and thus the relaxation will be more effective and relatively cheap on such grid [100]. Suppose that an approximation to the error e_h is obtained on Ω_{2h} through relaxation or any other suitable procedure (e.g. a direct method) then that approximation can be used to correct the fine grid solution (hence known as coarse grid correction). A combination of error smoothing and its coarse grid correction leads to so-called two-grid method [21, 100]. A two-grid method is mainly based on the following components [21, 35, 93, 100]:

- Smoother: a relaxation scheme which efficiently reduces the high-frequency error components through its first few iterations.
- Prolongation or interpolation: a transfer operator I_{2h}^h which maps a coarse grid vector v_{2h} to a fine grid vector v_h:

$$I_{2h}^h : v_{2h} \longrightarrow v_h$$

- Restriction: a transfer operator I_h^{2h} which defines a mapping from a fine grid to a coarse grid:

$$I_h^{2h} : v_h \longrightarrow v_{2h}$$

Having defined these components, a two-grid method can be summarized in the following steps:

- relax ν_1 times $A_h x_h = b_h$ on Ω_h with an initial guess for x_h
- restrict the residual: $r_h = b_h - A_h \tilde{x}_h$ to the coarse grid by $r_{2h} = I_h^{2h} r_h$
- solve the coarse grid problem:

$$A_{2h} e_{2h} = r_{2h} \tag{3.31}$$

- prolongate the coarse grid error e_{2h} to the fine grid by $e_h = I_{2h}^h e_{2h}$ and correct the fine grid approximation: $\tilde{x}_h \leftarrow \tilde{x}_h + e_h$
- relax ν_2 times $A_h x_h = b_h$ on Ω_h with the initial guess $\tilde{x}_h$.

The numbers ν_1 and ν_2 are also known as pre-and post-smoothing iterations respectively. In practice, the computational complexity of the coarse grid problem (3.31) will itself be very large therefore it is useful to solve it approximately. To obtain such an approximation, a further two-grid strategy can be applied to the problem (3.31). This process can be repeated recursively until a coarsest grid is reached for which a direct solution is possible at very low cost. Such a recursive application of a two-grid correction scheme leads to what is called a true "multigrid method" [21, 100]. An example of a sequence of subsequent coarse grids used in the multigrid are shown in Figure 3.2. The recursive definition of multigrid method is summarized in Algorithm 5.

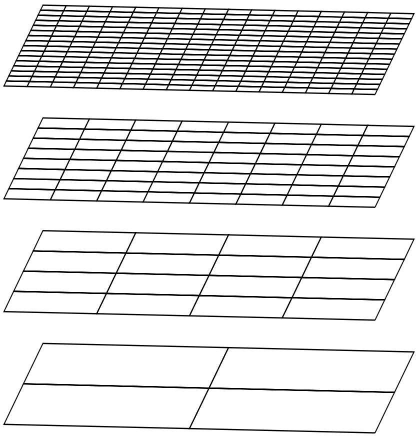


Figure 3.2: A sequence of hierarchical grids

The method outlined in Algorithm 5 is regarded as a 1 V-cycle of multigrid. A schematic of a multigrid V-cycle is also shown in Figure 3.3. The details about other

Algorithm 5 $\tilde{x}_h$=MGV(A_h, x_0, b_h) [21]

```
apply $\nu_1$ pre-smoothing steps on $A_h x_h = b_h$ with an initial guess $x_0$
restrict the residual: $r_h = b_h - A_h \tilde{x}_h$ to the coarse grid by $r_H = I_h^H r_h$
if coarsest grid is reached then
    solve: $A_H e_H = r_H$
else
    $e_H$=MGV$(A_H, 0, r_H)$
end if
correct the fine grid approximation: $\tilde{x}_h \leftarrow \tilde{x}_h + I_H^h e_H$
apply $\nu_2$ post-smoothing steps on $A_h x_h = b_h$ with the initial guess $\tilde{x}_h$
```

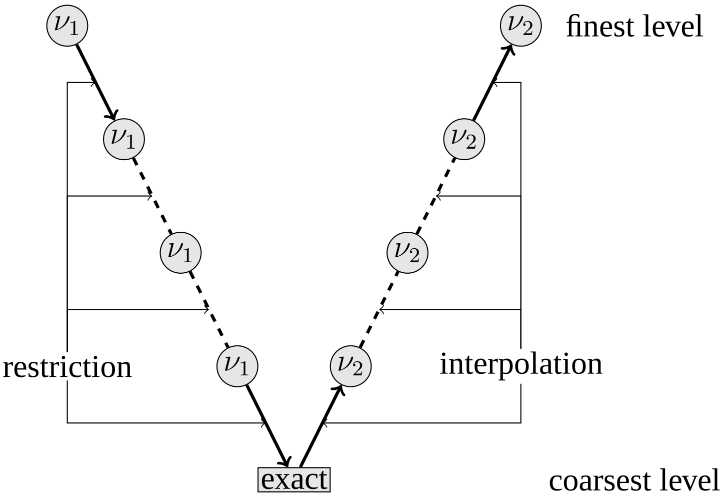


Figure 3.3: Schematic of a multigrid V-cycle

multigrid cycles can be found in the standard texts, for example, see [21, 100]. Note that the error of the improved solution may still be large, thus the process can be repeated until the solution satisfies some prescribed stopping tolerance. Multigrid methods are designed to have h-independent convergence and they therefore require only $\mathcal{O}(N)$ operations [100]. Instead of using multigrid as a stand-alone solver they can be combined with iterative methods such as conjugate gradient [44], GMRES [94] etc. to accelerate their convergence. In fact, a single V-cycle of the multigrid can be used as an efficient preconditioner (to approximately solve the preconditioned system: $Mz = r$) for an iterative solver with the computational complexity proportional to the size of problem. So this is the context in which a multigrid approach will be described, and used, here. The multigrid approach can be classified into two variants called geometric multigrid (GMG) and algebraic multigrid (AMG), and these are discussed in the following sections.

3.7.1 Geometric Multigrid

The geometric multigrid method [21, 100] (GMG) is explicitly tied to the underlying differential equation and the discretization scheme used. It depends on the availability of both discretization and underlying mesh information. This involves generating a sequence of hierarchical meshes $\Omega_1, \Omega_2, ..., \Omega_n$ such that $\Omega_1 \subset \Omega_2 \subset .....\Omega_n$. Such a hierarchy of grids is obtained through uniform or adaptive refinements of elements of a particular grid, where each selected element is automatically divided into smaller finer elements. Figure 3.2 shows a sequence of such 4-hierarchical grids with rectangular elements. This then requires the discretization of a problem at each individual grid level. Moreover the inter-grid transfer operators are defined using the geometric information.

3.7.2 Algebraic Multigrid

This is an extension of the multigrid approach where the construction of the multilevel hierarchy takes place algebraically, hence the name algebraic multigrid (AMG) [17, 21, 93, 100]. AMG is being considered to develop effective and robust preconditioners for Krylov subspace methods [17]. In AMG, the re-discretization of a problem on each coarse level and the geometric interpretation are not required to build the hierarchy of discrete systems. Therefore AMG has an advantage over the geometric multigrid regarding its ease in implementation, however this can also restrict the applicability of the approach too (e.g. the system matrix must satisfy certain AMG criteria [17]). An algebraic multigrid method builds the hierarchy of operators directly from the original system matrix. AMG can be distinguished into two phases: the setup phase and the solve phase. In the setup phase the hierarchy of linear systems is developed along with the transfer operators: the prolongation operator I_{coarse}^{fine} and the restriction operator $I_{fine}^{coarse} = (I_{coarse}^{fine})^T$. The Galerkin condition is usually used to construct the coarse-grid operator [17, 21, 35]:

$$A^{coarse} = I_{fine}^{coarse} A^{fine} I_{coarse}^{fine}.$$

The second phase is regarded as the solution phase where the standard multigrid cycles are performed (typically a single V-cycle is used for preconditioning). The unknowns of subsequent coarse levels are simply the subsets of the original unknowns of the problem. This is based on finding the strength of connections from the entries of a system matrix [100]. Note that a point i has a strong connection with point j if:

$$-a_{ij} = \theta \max_{k \neq i}\{-a_{ik}\}, \qquad \theta \in (0, 1]$$

i.e the unknown x_i strongly depends on the unknown x_j or the unknown x_j strongly influences the unknown x_i. This eventually leads to partitioning into a set of coarse points C and fine points F. Once the coarse grid points have been chosen then the interpolation weights are defined. Having defined the prolongation operator, remaining operators are defined as stated above.

The standard coarsening is usually performed in two passes of the data. In the first pass, a preliminary partitioning into C and F unknowns is done. In the second, some of the initial F unknowns are reassigned as C unknowns to further satisfy the prolongation requirements. If the second pass is not performed, the coarsening strategy is referred to as one or single pass coarsening [17, 21]. Single pass coarsening can have a significant effect on the setup phase where it not only reduces the time required by coarsening process but also reduces the storage requirement for the coarse grid operators. It has a further significant effect on the solution phase i.e it reduces the overall time required to perform a multigrid cycle. However a drawback is that it may result in a poorer quality preconditioner [17].

Throughout this thesis an open source software `HSL_MI20` [1, 17] has been used for the AMG preconditioning. AMG preconditioning [17] requires that the system matrix must have positive diagonal entries and most of the off-diagonal entries must be negative (the diagonal should be large compared to the sum of the off-diagonals). This software offers the flexibility of user-selected choices including the choice of smoother, coarsest grid solver and the choice of the type of coarsening. Note that one pass coarsening is assumed throughout this thesis if not explicitly stated otherwise.

3.8 KINSOL Implementation

The KINSOL (Krylov Inexact Newton Solver) software is a member of the family of the SUNDIALS software suite [57] developed to solve systems of nonlinear algebraic equations. This is a C implementation of previous software NKSOL [23], a Newton-Krylov solver written in FORTRAN to solve systems of nonlinear algebraic equations. However KINSOL provides some additional features including a wider choice of linear system solvers and tolerances. The linear solver modules available within KINSOL can be categorized into two families: direct family and “spils” family. The former offers two direct linear solvers for dense and banded system matrices, while the latter comprises of three scaled preconditioned iterative linear solvers which are based on Krylov subspace methods. These are respectively scaled preconditioned GMRES [94] (SPGMR), a scaled preconditioned Bi-Conjugate Gradient Stable method [101] (SPBCG) and a scaled preconditioned Transpose-Free Quasi-Minimal Residual method [43] (SPTFQMR). In addition to these modules, KINSOL also provides an option of a user-own linear solver module. KINSOL implements a modified Newton strategy with a direct linear solver where the Jacobian updates are made as infrequently as possible [57] to balance the overall high computational costs. However the drawback of this method is that it requires more nonlinear iterations to converge. In the case of spils solvers, KINSOL implements an inexact Newton strategy. For the Krylov iterative solvers, right preconditioning is available within KINSOL. In this case KINSOL requires user subroutines to setup and solve the preconditioned system of the form $Mz = r$ where M denotes a preconditioning matrix. There are two options to choose the length of Newton step (λ): in the first case λ is set to 1 (standard Newton strategy) while the second option (global strategy) uses a linesearch strategy to determine a suitable length of Newton step [57, 68].

In KINSOL the stopping criteria for the Newton method are based upon both the nonlinear residual and the step length. For the former, the Newton iteration must pass a stopping test:

$$\|F(u_n)\|_\infty < U^{\frac{1}{3}},$$

where U is the machine unit roundoff. For the latter, the Newton method will terminate when the maximum scaled step is below a given tolerance

$$\|\lambda\delta_n\|_\infty < U^{\frac{2}{3}}.$$

Only the first condition is considered a successful completion of KINSOL. The second condition (small step) may indicate that the iteration is stalled near a point for which the residual is still unacceptable. Note that the above mentioned values are defaults however KINSOL also accepts the user's own specified values for these tolerances.

In the **inexact Newton strategy**, i.e. when the preconditioned iterative solver is used, the convergence of the overall nonlinear solver is coupled with the accuracy with which the linear system is solved at each Newton iteration. Specifically, the stopping criterion for the Krylov iteration is chosen as:

$$\|F'(u_n)\,\delta_n + F(u_n)\| < (\eta_n + U)\|F(u_n)\|\ .$$

Different choices of η_n are available within KINSOL, including the user's own choice or a default value of 0.1.

Finally the system Jacobian can be supplied to KINSOL by a user subroutine or it can be approximated within KINSOL using the difference quotient [68]. Since Krylov iterative solvers require Jacobian-vector multiplication, the difference quotient approximation of Jacobian-vector product avoids the storage of Jacobian matrix thus a significant amount of memory may be saved particularly for large systems.

Throughout this thesis KINSOL is applied using the SPGMR module for the preconditioned iterative solutions if not explicitly stated otherwise. The SPGMR module uses the GMRES method with no restarts. User subroutines have been provided to calculate Jacobian-vector products at each inner iteration and to solve the preconditioning systems. For the direct solutions, an external linear solver (UMFPACK [28]) is attached within KINSOL.

3.9 Summary

The content of this chapter reflects the numerical techniques which have been used in the rest of this thesis. First of all, the finite element method has been explained for the solution of partial differential equations. Later on, various numerical techniques have been discussed for the solution of linear and nonlinear systems of equations. These methods can be implemented within the framework of the KINSOL library [57], which has been used throughout this work to develop codes for the solution of EHL equations. KINSOL allows the choice of inner solvers: both direct and preconditioned iterative solvers. For the direct inner-solver, UMFPACK [28] can be attached for inner direct solves. KINSOL implements an inexact Newton strategy with a preconditioned iterative linear solver. Three scaled preconditioned linear iterative solvers: GMRES [94], Bi-CGStab [101] and TFQMR [43] are available within KINSOL through the modules SPGMR, SPBCG and SPTFQMR respectively [57]. The SPGMR module with no scaling is used throughout this thesis if not explicitly stated otherwise. Finally a user subroutine may be supplied to KINSOL to solve the preconditioned system at each linear iteration. Moreover a standard Newton strategy, i.e. a full step, is taken to update the solution at each Newton iteration.

Chapter 4

Discretization and Solution

4.1 Overview

In this chapter the full-system finite element approach to the solution of EHL equation is discussed. In the following section we described a standard Galerkin finite element discretization of EHL line and point contact equations. Later on, an alternative discretization of the Reynolds equation is discussed in order to stabilize the pressure solution. This is then followed by an explanation of the coupling procedure and a description of the Newton method applied to the fully-coupled nonlinear system. Later in this chapter a new preconditioning strategy is proposed and discussed for the development of an efficient iterative solver for the resulting linear systems at each Newton step. Following a consideration of different variants of the solver used in this work, the accuracy of the solution is justified both for the line and the point cases compared to previously published results.

4.2 Finite Element Discretization

In the previous chapter, a general finite element procedure was introduced for the Poisson equation. Now in this section the finite element discretization of the EHL equations is discussed both for the line and the point contact cases. Initially we consider a standard Galerkin discretization but, as the Reynolds equation is convection-dominated in the contact region it exhibits oscillations in its pressure solution [73, 102]. Therefore in

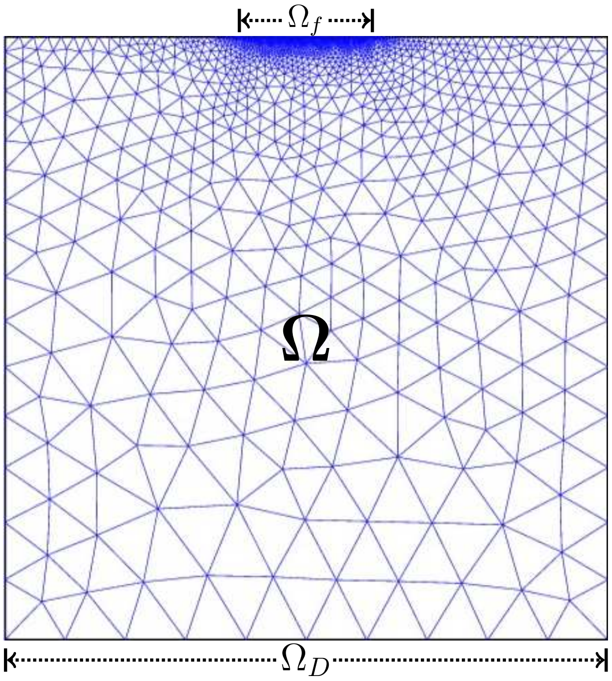


Figure 4.1: A 2D domain and a triangulation

Section 4.2.3 an alternative discretization of the Reynolds equation is discussed. This is shown to be more stable in the sense that the unphysical pressure oscillations no longer appear.

4.2.1 Line Contact

In this subsection, standard Galerkin discretizations of the Reynolds equation (in Ω_f) and the linear elasticity equation (in Ω) are considered. Recall from Chapter 2 that the line contact problems requires the Reynolds equation to be solved on a one dimensional domain Ω_f for pressure distribution, and the linear elasticity equation on a two dimensional domain Ω for the elastic deformation solution. Let $\partial\Omega$ be the boundary of the domain Ω and Ω_f be the part of $\partial\Omega$ that corresponds to the fluid region, see Figure 4.1. For the line contact case the penalised Reynolds equation (2.29) takes the following form:

$$\frac{\mathrm{d}}{\mathrm{d}X}\left(\epsilon(P)\frac{\mathrm{d}P}{\mathrm{d}X}\right) - \frac{\mathrm{d}}{\mathrm{d}X}\left(\bar{\rho}H\right) - \xi P^{-} = 0, \tag{4.1}$$

where $\epsilon(P) = \frac{\bar{\rho}H^3}{\bar{\eta}\lambda}$ and $P = 0$ on $\partial\Omega_f$. For a weighting function W_p (which is zero on $\partial\Omega_f$), the weak form of this equation is:

$$\int_{\Omega_f} \epsilon(P)\frac{\mathrm{d}P}{\mathrm{d}X}\frac{\mathrm{d}W_p}{\mathrm{d}X}\mathrm{d}\Omega_f - \int_{\Omega_f} \bar{\rho}H\frac{\mathrm{d}W_p}{\mathrm{d}X}\mathrm{d}\Omega_f + \xi\int_{\Omega_f} P^- W_p \mathrm{d}\Omega_f = 0. \tag{4.2}$$

Similarly, for a weighting function W_u (which has two components, both of which are zero on Ω_D), the weak form of the linear elasticity equation (2.34) is:

$$\int_{\Omega} C_{ijkl}\frac{\partial U_k}{\partial X_l}\frac{\partial W_u}{\partial X_j}d\Omega = \int_{\Gamma_t} n_j C_{ijkl}\frac{\partial U_k}{\partial X_l}W_u d\Gamma_t, \tag{4.3}$$

where $\Gamma_t (= \partial\Omega - \Omega_D)$ is the part of $\partial\Omega$ that corresponds to the traction boundary. As shown in equation (2.35) $n_j C_{ijkl}\frac{\partial U_k}{\partial X_l}$ represents the normal traction and is given as

$$\begin{cases} n_j C_{ijkl}\frac{\partial U_k}{\partial X_l} &= 0 \qquad (\text{for } i = 1, 2) \qquad \text{on } \Gamma_t - \Omega_f, \\ n_j C_{ijkl}\frac{\partial U_k}{\partial X_l} &= -\delta_{i2}P \qquad\qquad\qquad\quad \text{on } \Omega_f. \end{cases}$$

Equation (4.3) can therefore be written as:

$$\int_{\Omega} C_{ijkl}\frac{\partial U_k}{\partial X_l}\frac{\partial W_u}{\partial X_j}d\Omega + \int_{\Omega_f} \delta_{i2}PW_u d\Omega_f = 0. \tag{4.4}$$

Consider a partition of Ω into ne_u triangular elements, $\{\Omega_e\}$, such that ne_p of these elements have edges on Ω_f (and these edges form a partition of Ω_f). Let $\{\Omega_{fe}\}$ be the set of these ne_p line segments (edges) and define $\mathcal{P}_u^h \subset H_0^1$ and $\mathcal{P}_p^h \subset H_0^1$ to be piecewise linear finite element solution spaces over $\{\Omega_e\}$ and $\{\Omega_{fe}\}$ respectively. Equations (4.2) and (4.4) can be written as a sum over these elements:

$$\sum_{\Omega_{fe}=1}^{ne_p}\left(\int_{\Omega_{fe}} \epsilon(P)\frac{\mathrm{d}P}{\mathrm{d}X}.\frac{\mathrm{d}W_p}{\mathrm{d}X}\mathrm{d}\Omega_{fe} - \int_{\Omega_{fe}} \bar{\rho}H\frac{\mathrm{d}W_p}{\mathrm{d}X}\mathrm{d}\Omega_{fe} + \xi\int_{\Omega_{fe}} P^- W_p \mathrm{d}\Omega_{fe}\right) = 0, \tag{4.5}$$

$$\sum_{\Omega_e=1}^{ne_u}\int_{\Omega_e} C_{ijkl}\frac{\partial U_k}{\partial X_l}\frac{\partial W_u}{\partial X_j}d\Omega_e + \sum_{\Omega_{fe}=1}^{ne_p}\int_{\Omega_{fe}} \delta_{i2}PW_u d\Omega_{fe} = 0. \tag{4.6}$$

On each element Ω_{fe}, P in equation (4.5) may be approximated by:

$$P \approx P^e = \sum_{j=1}^{2} N_j^e P_j^e, \tag{4.7}$$

where N_j^e are local linear basis functions and P_j^e are nodal values of P at the vertices of the element. So replacing P in equation (4.5) with the approximation given in equation (4.7) and W_p with N_i^e for $i = 1, 2$, followed by the usual finite element assembly process, leads to a discrete system of n_p nonlinear equations, where n_p is total number of nodes in Ω_f, excluding boundary nodes. Note that there is no explicit requirement for the meshes of the domains Ω and Ω_f to match. However numerical experience [98] suggests that a similar resolution is required in both Ω and Ω_f, thus we have used a single mesh in this work for ease of implementation.

Similarly, on each element Ω_e, U in equation (4.6) is approximated by:

$$U \approx U^e = \sum_{q=1}^{3} W_q^e U_q^e, \tag{4.8}$$

where W_q^e are local linear basis functions and U_q^e are nodal values of the displacement U within the element. So replacing U in equation (4.6) with the approximation given in equation (4.8), P with the approximation given in equation (4.7) and W_u with $(W_q^e, 0)^T$ and then $(0, W_q^e)^T$ for $q = 1, 2, 3$, leads (after finite element assembly) to a discrete system of $2 \times n_u$ linear equations, where n_u is total number of nodes in Ω excluding Dirichlet boundary nodes on Ω_D.

Note that the choice of the linear elements discussed above leads to highly sparse problems and, as we shall see next, preconditioning techniques such as algebraic multigrid are very appropriate for linear elements. The price that is paid for using these elements however is their lower accuracy (than higher degree approximations) which means that much finer meshes are required that might be the case for higher order elements.

Finally, the discrete form of the load balance equation (2.19) is obtained by using the finite element assembly of the approximation given in (4.7):

$$\sum_{\Omega_{fe}=1}^{ne_p} \int_{\Omega_{fe}} \sum_{j=1}^{2} N_j^e P_j^e d\Omega_{fe} - \frac{\pi}{2} = 0. \tag{4.9}$$

4.2.2 Point Contact

This subsection considers a standard Galerkin discretization of the Reynolds equation (in Ω_f) and the linear elasticity equation (in Ω) for the point contact case. Here the Reynolds equation needs to be solved on a two dimensional domain Ω_f, and the elasticity equation on a three dimensional domain Ω. Again we define that $\partial\Omega$ is the boundary of the domain Ω and Ω_f is the part of $\partial\Omega$ that corresponds to the fluid region. The penalised Reynolds equation (2.29) for the point contact case reads:

$$\nabla.\left(\epsilon\nabla P\right)-\frac{\partial}{\partial X}\left(\bar{\rho}H\right)-\xi P^{-}=0, \tag{4.10}$$

where $P=0$ around $\partial\Omega_f$. For a weighting function W_p ($W_p=0$ on $\partial\Omega_f$), the weak form of the equation (4.10) is:

$$\int_{\Omega_f}\epsilon\nabla P.\nabla W_p d\Omega_f-\int_{\Omega_f}\bar{\rho}H\frac{\partial W_p}{\partial X}d\Omega_f+\xi\int_{\Omega_f}P^{-}W_p d\Omega_f=0. \tag{4.11}$$

Similarly, for a weighting function W_u ($W_u=\mathbf{0}$ on Ω_D), the weak form of the linear elasticity equation is:

$$\int_{\Omega}C_{ijkl}\frac{\partial U_k}{\partial X_l}\frac{\partial W_u}{\partial X_j}d\Omega=\int_{\Gamma_t}n_j C_{ijkl}\frac{\partial U_k}{\partial X_l}W_u d\Gamma_t, \tag{4.12}$$

where $\Gamma_t(=\partial\Omega-\Omega_D)$ is the part of $\partial\Omega$ that corresponds to the traction boundary. As for the line contact case $n_j C_{ijkl}\frac{\partial U_k}{\partial X_l}$ represents the normal traction and for the point contact case, it is given by

$$\begin{cases} n_j C_{ijkl}\frac{\partial U_k}{\partial X_l} &= 0 \quad (\text{for } i=1,2,3) \qquad \text{on } \Gamma_t-\Omega_f, \\ n_j C_{ijkl}\frac{\partial U_k}{\partial X_l} &= -\delta_{i3}P \qquad\qquad\qquad\quad \text{on } \Omega_f. \end{cases}$$

Equation (4.12) can therefore be written as:

$$\int_{\Omega}C_{ijkl}\frac{\partial U_k}{\partial X_l}\frac{\partial W_u}{\partial X_j}d\Omega+\int_{\Omega_f}\delta_{i3}PW_u d\Omega_f=0. \tag{4.13}$$

Now consider a partition of Ω into ne_u tetrahedral elements, $\{\Omega_e\}$, such that ne_p of these elements have triangular faces on Ω_f (and these triangular faces form a partition of Ω_f). Let $\{\Omega_{fe}\}$ be the set of these ne_p triangles and define $\mathcal{P}_u^h$ and $\mathcal{P}_p^h$ to be piecewise linear finite element spaces over $\{\Omega_e\}$ and $\{\Omega_{fe}\}$ respectively. Equations (4.11) and (4.13) can be written as a sum over these elements:

$$\sum_{\Omega_{fe}=1}^{ne_p}\left(\int_{\Omega_{fe}}\epsilon\nabla P.\nabla W_p d\Omega_{fe}-\int_{\Omega_{fe}}\bar{\rho}H\frac{dW_p}{dX}d\Omega_{fe}+\xi\int_{\Omega_{fe}}P^{-}W_p d\Omega_{fe}\right)=0, \tag{4.14}$$

$$\sum_{\Omega_e=1}^{ne_u}\int_{\Omega_e}C_{ijkl}\frac{\partial U_k}{\partial X_l}\frac{\partial W_u}{\partial X_j}d\Omega_e+\sum_{\Omega_{fe}=1}^{ne_p}\int_{\Omega_{fe}}\delta_{i3}PW_u d\Omega_{fe}=0. \tag{4.15}$$

On each element Ω_{fe}, P in equation (4.14) is approximated by:

$$P \approx P^e = \sum_{j=1}^{3} N_j^e P_j^e, \tag{4.16}$$

where N_j^e are local linear basis functions and P_j^e are nodal values of P within the element. So replacing P in equation (4.14) with the approximation given in equation (4.16) and W_p with N_i^e for $i = 1, 2, 3$, followed by the usual finite element assembly process, leads to a discrete system of n_p nonlinear equations, where n_p is total number of nodes in Ω_f, excluding boundary nodes. Similarly, on each element Ω_e, U in equation (4.15) is approximated by:

$$U \approx U^e = \sum_{q=1}^{4} W_q^e U_q^e, \tag{4.17}$$

where W_q^e are local linear basis functions and U_q^e are nodal values of the displacement U within the element. So replacing U in equation (4.15) with the approximation given in equation (4.17), P with the approximation given in equation (4.16) and W_u with $(W_q^e, 0, 0)^T$, $(0, W_q^e, 0)^T$ and then $(0, 0, W_q^e)^T$ for $q = 1, ..4$, leads (after finite element assembly) to a discrete system of $3 \times n_u$ linear equations, where n_u is total number of nodes in Ω excluding Dirichlet boundary nodes.

Finally, the discrete form of the load balance equation is obtained by using the finite element assembly of the approximation given in (4.16):

$$\sum_{\Omega_{fe}=1}^{ne_p} \int_{\Omega_{fe}} \sum_{j=1}^{3} N_j^e P_j^e d\Omega_{fe} - \frac{2\pi}{3} = 0. \tag{4.18}$$

4.2.3 Stabilization

For heavily loaded cases the value of diffusivity, $\epsilon(P) = \frac{\bar{\rho} H^3}{\bar{\eta}\lambda}$, becomes very small in the contact region due to large increases in the viscosity, $\bar{\eta}$, of the lubricant used, and so the Reynolds equation is convection-dominated in this region. In this situation the Galerkin solution exhibits an oscillatory behaviour in the pressure, see for example [73, 102]. For such cases, a standard Galerkin approximation of the Reynolds equation (discussed in the previous section) is unsatisfactory [51]. In order to obtain a stabilized solution a Streamline Upwind Petrov-Galerkin (SUPG) method [22] may be used. For the sake of brevity, the implementation is explained for the point contact case only (the line contact case being a simplification of this).

To describe the SUPG approach let us rewrite equation (4.10) (excluding penalty term) in the classical convection-diffusion form as follows:

$$-\nabla.\,(\epsilon\nabla P) + H\frac{\partial\bar{\rho}}{\partial P}\frac{\partial P}{\partial X} + \bar{\rho}\frac{\partial H}{\partial X} = 0,$$

or

$$-\nabla.\,(\epsilon\nabla P) + V.\nabla P + Q = 0, \tag{4.19}$$

where $V = (V_1, V_2) = (H\frac{\partial\bar{\rho}}{\partial P}, 0)$ and $Q = \bar{\rho}\frac{\partial H}{\partial X}$. The SUPG method applied to this kind of problem consists of choosing the weighting function W_p in equation (4.14) to be of the form:

$$W_p = N_p + \alpha\frac{h_e}{2|V|}V.\nabla N_p$$

with

$$\alpha = \coth(Pe) - \frac{1}{Pe}, \qquad Pe = \frac{|V|h_e}{2\epsilon},$$

where h_e is the element size and Pe is the local Peclet number of the element e. In the Galerkin procedure, oscillations will occur whenever $|Pe| > 1$. For more details, see for example [22, 35, 51, 108].

Note that if the discretization procedure given in Section 4.2.2 when applied to equation (4.19) with the new weighting function defined above end up with the following form:

$$\sum_{\Omega_{fe}=1}^{ne_p}\left(\int_{\Omega_{fe}}\epsilon\nabla P.\nabla N_p d\Omega_{fe} - \int_{\Omega_{fe}}\bar{\rho}H\frac{dN_p}{dX}d\Omega_{fe} + \alpha\frac{h_e}{2|V|}\int_{\Omega_{fe}}(V.\nabla N_p)\frac{d(\bar{\rho}H)}{dX}d\Omega_{fe}\right) = 0, \tag{4.20}$$

So replacing P in equation (4.20) with the approximation given in equation (4.16) and N_p with N_i^e for $i = 1, 2, 3$, followed by the usual finite element assembly process, leads to a discrete system of n_p nonlinear equations, where n_p is total number of nodes in Ω_f, excluding boundary nodes. This method introduces an additional stabilization term within the Reynolds equation which is mesh dependent, therefore it has no effect on the accuracy of a converged solution in the limit as the mesh size goes to zero.

4.3 Solution Method

4.3.1 Coupling Procedure

The traditional half-space approaches are mainly based on a loose-coupling of the pressure and film thickness solutions. In such an approach (e.g. [20, 46, 47, 73, 102, 104]), each EHL equation is solved separately and an iterative link is established between their solutions. Typically under-relaxation is required to achieve convergence of the solution and consequently a slow convergence rate may be obtained. A further decrease in under-relaxation factor is required to achieve the convergence for heavily loaded cases, which leads to a further decrease in the convergence rate [104].

In the fully coupled approach, all the discrete systems arising from the FEM discretization of EHL equations are coupled together to form a large nonlinear system of equations for all of the unknowns, and this is solved in a single pass. Thus, no under-relaxation is required and no extra treatment is required to achieve convergence for heavily loaded cases (e.g. [50, 51]). Nevertheless, some heavily loaded cases may require a better initial guess for a Newton procedure to achieve convergence.

4.3.2 Linearization

In Section 4.2 we discussed the discrete nonlinear systems arising from EHL equations. Let us rewrite them in the following vector form:

$$\begin{cases} \mathbf{R}_P(\mathbf{P}, \mathbf{U}, H_0) & = \mathbf{0} \\ \mathbf{R}_U(\mathbf{P}, \mathbf{U}) & = \mathbf{0} \\ R_{H_0}(\mathbf{P}) & = 0 \quad , \end{cases} \tag{4.21}$$

where $\mathbf{R}_P$ represents the residual of the system of nonlinear equations arising from the discretization of Reynolds equation, $\mathbf{R}_U$ is the residual of the linear system of equations arising from discretization of the linear elasticity equation and R_{H_0} is the residual of the discretized load balance equation. In this work, a Newton procedure is applied to system (4.21) to yield the following linear system at each outer iteration:

$$\begin{bmatrix} \frac{\partial \mathbf{R}_P}{\partial \mathbf{P}} & \frac{\partial \mathbf{R}_P}{\partial \mathbf{U}} & \frac{\partial \mathbf{R}_P}{\partial H_0} \\ \frac{\partial \mathbf{R}_U}{\partial \mathbf{P}} & \frac{\partial \mathbf{R}_U}{\partial \mathbf{U}} & \mathbf{0} \\ \frac{\partial R_{H_0}}{\partial \mathbf{P}} & \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{P} \\ \delta \mathbf{U} \\ \delta H_0 \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_{\mathbf{P}} \\ -\mathbf{R}_{\mathbf{U}} \\ -R_{H_0} \end{bmatrix} . \tag{4.22}$$

Starting with an initial estimate for the solution, the Newton procedure consists of solving the linearized system (4.22) at each Newton iteration and this update is added to the solution obtained at the previous iteration, to provide an updated solution. This process is repeated until convergence is achieved.

4.3.3 Structure of the Linear System

In what follows we write the Jacobian matrix in the system (4.22) in the following form:

$$\mathbf{J} = \left[\begin{array}{ccc} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & 0 \\ J_{31} & 0 & 0 \end{array}\right], \tag{4.23}$$

where J_{11} is a sparse block (tridiagonal in the line contact case) of size $n_p \times n_p$. Also $J_{12},\ J_{21}$ and J_{22} are highly sparse blocks of size $n_p \times (d \times n_u)$, $(d \times n_u) \times n_p$ and $(d \times n_u) \times (d \times n_u)$ respectively, where d is 2 or 3 for line or point contact problems respectively. Finally, J_{13} and J_{31} are column and row vectors, respectively, of length n_p. Note that $n_p \ll n_u$ since the Reynolds equation is solved in a domain of dimension $d-1$.

4.4 Preconditioned Iterative Solution

As described in the previous section the Newton method requires the solution of the linear system (4.22) at each iteration which is the most expensive part of a Newton iteration. Therefore one needs to solve this system as efficiently as possible in order to get the best performance results. Both direct and iterative approaches can be used to solve the linear system (4.22) at each Newton step. The objective of this study is to develop an efficient preconditioned iterative solver so that the system (4.22) can be solved efficiently (in both time and memory) at each Newton step, as compared to a state-of-the-art sparse direct solver (e.g. [28]). This goal can be achieved if a good preconditioner is available at a relatively low cost. A simplest choice of the preconditioner may be a block diagonal preconditioner of the form:

$$\mathbf{P} = \left[\begin{array}{ccc} \widetilde{J}_{11} & 0 & 0 \\ 0 & \widetilde{J}_{22} & 0 \\ 0 & 0 & 1 \end{array}\right],$$

where $\widetilde{J}_{11} \approx J_{11}$ and $\widetilde{J}_{22} \approx J_{22}$ in some sense. This preconditioner can be used to precondition the Reynolds and the elasticity block in system matrix (4.23) separately. In order to get a better preconditioner, consider, for simplicity, the case where the force balance equation is ignored. Then we have a (2×2) block Jacobian matrix of the form

$$\hat{\mathbf{J}} = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}$$

If we choose a left preconditioner $\hat{\mathbf{P}}$ such that

$$\hat{\mathbf{P}}^{-1} = \begin{pmatrix} S^{-1} & -S^{-1}J_{12}J_{22}^{-1} \\ 0 & J_{22}^{-1} \end{pmatrix}$$

then

$$\hat{\mathbf{P}}^{-1}\hat{\mathbf{J}} = \begin{pmatrix} S^{-1} & -S^{-1}J_{12}J_{22}^{-1} \\ 0 & J_{22}^{-1} \end{pmatrix} \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} = \begin{pmatrix} S^{-1}(J_{11} - J_{12}J_{22}^{-1}J_{21}) & 0 \\ J_{22}^{-1}J_{21} & I \end{pmatrix}.$$

For an ideal preconditioner

$$S = J_{11} - J_{12}J_{22}^{-1}J_{21},$$

i.e. all eigenvalues of $\hat{\mathbf{P}}^{-1}\hat{\mathbf{J}}$ are equal to 1, thus the convergence should be obtained in precisely 2 iterations [77]. However, the question arises as to how cheaply the effect of S^{-1} can be computed (or approximated) without forming S explicitly. It should be noted that, since $n_p \ll n_u$, even with a relatively poor approximation of the Schur complement S, still the vast majority of the eigenvalues of $\hat{\mathbf{P}}^{-1}\hat{\mathbf{J}}$ will be equal to 1. Furthermore, it will be shown that the application of this method with $S = J_{11}$ yields a preconditioner that is both cost-effective and highly efficient in accelerating the convergence of GMRES [94].

In this work, an identity "preconditioner" is used for the single load balance equation. Thus the preconditioner (to solve $\mathbf{Pz} = \mathbf{r}$, at each GMRES iteration) used here is based upon the following upper triangular form:

$$\mathbf{P} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ 0 & \widetilde{J}_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In the case of line contact problems, J_{11} is a tridiagonal matrix, so the effect of J_{11}^{-1} over an arbitrary vector ν is calculated efficiently using the Thomas algorithm which is $\mathcal{O}(n_p)$. The block preconditioner, $\widetilde{J}_{22} \approx J_{22}$, of the elasticity block is undertaken by either algebraic multigrid (AMG) preconditioning [1, 17] or geometric multigrid (GMG) preconditioning [21, 100], which are designed to be $\mathcal{O}(n_u)$. In the case of point contact problems, J_{11} is no longer tridiagonal, but is still a highly sparse (and relatively small) block, so a sparse direct solver [28] is used to compute the effect of J_{11}^{-1} over an arbitrary vector ν.

The overall algorithm to solve the preconditioned system:

$$\begin{bmatrix} J_{11} & J_{12} & J_{13} \\ 0 & \widetilde{J}_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{z_P} \\ \mathbf{z_U} \\ z_{H_0} \end{bmatrix} = \begin{bmatrix} \mathbf{r_P} \\ \mathbf{r_U} \\ r_{H_0} \end{bmatrix} \tag{4.24}$$

therefore involves the following steps:

1. Form $z_{H_0} = r_{H_0}$

2. Perform 1-V cycle of AMG or GMG to compute $\widetilde{J}_{22}^{-1}\mathbf{r_U}$, which is the approximate solution of $J_{22}\mathbf{z_U} = \mathbf{r_U}$.

3. Using the calculated solutions in step-1 & step-2, solve:

$$J_{11}\mathbf{z_P} = \mathbf{r_P} - J_{12}\mathbf{z_U} - J_{13}z_{H_0}$$

   for $\mathbf{z_P}$ using the Thomas algorithm in the line contact case or a sparse direct solver [28] in the point contact case.

It should be noted that the application of AMG preconditioning [1, 17] requires that the system matrix must have positive diagonal entries and most of the off-diagonal entries must be negative (the diagonal should be large compared to the sum of the off-diagonals). As described above, the size of the J_{22} block is $(d \times n_u) \times (d \times n_u)$, which corresponds to $d \times n_u$ displacement unknowns with n_u unknowns in each direction. If we order the unknown displacements for each coordinate direction in turn, then the J_{22} block is represented by d^2 sub-blocks of sizes $(n_u \times n_u)$. In order to apply AMG preconditioning to J_{22} we apply a single V-cycle to each of the diagonal sub-blocks, and neglect the off-diagonal blocks. Finally, a single pass coarsening is used in AMG preconditioning. This strategy leads to fewer and much sparser coarse level operators and therefore results in a significant reduction in the memory usage and overall time. Moreover, a Gauss-Seidel smoother is used both in AMG and GMG preconditioning. This approach to preconditioning the linear elasticity equation has been found to be very effective provided the Poisson ratio is not close to 0.5 [16]. In this work a Poisson ratio 0.3 has typically been used, however we assess the effect of altering this value in Chapter 5. The equivalent Poisson ratio and the Young's modulus are then obtained according to equations (2.36) and (2.37) for the reduced system models considered in this work.

4.5 Solver Layout

The main objective of this study is to propose a robust preconditioned iterative solver which is both computationally and memory efficient compared to a state-of-the-art sparse direct solver [27–30]. For this purpose, a nonlinear solver is developed in 'C', using the KINSOL [57] library to solve the nonlinear system (4.21). This nonlinear solver uses a standard Newton strategy, i.e it employs a full Newton step. Further, the stopping criteria for the Newton method is set to use the default values (see Section 3.8), if not explicitly stated otherwise. KINSOL requires a user subroutine to compute the system function (residual) of the nonlinear system (4.21) for a given value of current solution. The system Jacobian is computed and stored at the same time while computing the system function F. For the solution of the linearized system (4.22) at each Newton iteration, two variants of the solver are considered. The difference in these two variants is the attachment of different inner linear solvers within KINSOL for the solution of the linearized system (4.22) at each Newton iteration.

- **Sparse Direct Solver:** In the first variant, an external sparse direct solver is attached as an inner solver within KINSOL. Experience shows that UMFPACK [28] is a very efficient choice, therefore, this is used in this work.

- **Preconditioned Iterative Solvers:** The second variant of the nonlinear solver considers an iterative linear routine as the inner solver. If not explicitly stated otherwise, this is the preconditioned GMRES method [94] (without restarts) available within KINSOL through the SPGMR module [57]. For the stopping tolerance: $(\eta_n + U)||F||$ to terminate the GMRES iteration the value of η_n is supplied to KINSOL. The choice of η_n is highly empirical, and varies for the type of loaded cases being considered and the type of initial guess. The input initial guess is the Hertzian pressure profile for pressure solution and a positive number for H_0. With this poor initial guess we have found that $\eta_n = 10^{-6}$ works well in practice for the EHL problems that we have considered. Any significant increase in this parameter may lead to divergence of the solution as one can see that in the first few Newton iterations, where $||F||$ may be large, a bigger choice of this parameter may not lead to a good Newton step, and thus the Newton iteration may diverge. Furthermore, some heavily loaded EHL cases may require a further decrease in this parameter. A much smaller value of this parameter can suffer with some drawbacks however. One drawback may be that as the Newton iterate progresses the decrease in $||F||$ leads to successive drops in the linear solver tolerance and hence oversolving may cause the computational work to increase. Another drawback is that the convergence of the GMRES method may stall, especially, in the final few Newton iterations due to roundoff errors of double precision arithmetic. To avoid such situation the maximum dimension of Krylov subspace used can be fixed to a suitable constant in order to prevent the linear system from being oversolved, and the solution thus obtained (even if the maximum dimension of the Krylov subspace used is reached) can be used to update the Newton iterate. Another possible way is to choose η_n in such a way that the linear solver tolerance remains fixed to a suitable value unless the $\|F\|$ is not sufficiently small. In other words the inexact Newton strategy is only implemented in the final few Newton iterations. Such a treatment will be explicitly stated in the forthcoming text in Chapter 5.

A user subroutine is supplied to KINSOL for the evaluation of the Jacobian-vector product Jv for a given input vector v where the matrix J is the system Jacobian already available. Recall from Section 3.8 that only right preconditioning is available for the linear iterative solvers within KINSOL. A user subroutine is also supplied to solve the preconditioning system $\mathbf{P}\mathbf{z} = \mathbf{r}$, i.e. (4.24), at each GMRES iteration, where different blocks of the preconditioning matrix $\mathbf{P}$ were already computed along with the system Jacobian. The algorithm described in the previous section is used to solve the preconditioned system $\mathbf{P}\mathbf{z} = \mathbf{r}$ at each GMRES iteration. Finally, the preconditioned iterative variant of solver is split into **further two variants** on the basis of AMG and GMG preconditioning of $\widetilde{J}_{22}^{-1}\mathbf{z}_{\mathbf{U}} = \mathbf{r}_{\mathbf{U}}$ block in the preconditioned system (4.24).

4.6 Accuracy of EHL Solution

In this section we demonstrate the accuracy of the fully-coupled line and point contact solvers developed in this work. For this purpose the computed solutions are compared against previously published results using the integral approach (based upon a half-space formulation) in a finite difference based model. This comparison is discussed separately both for line and point contact problems in the following subsections.

4.6.1 Line Contact

In this subsection, a comparison of the fully-coupled finite element line contact results is established against a finite difference based model [102, 103] in order to validate the implementation for the test cases given in Table 4.1. We first establish this comparison

Table 4.1: Non-dimensional parameters for different line contact cases.

Parameters	Test Case 1	Test Case 2
Moes parameter, L	10	10
Moes parameter, M	20	200
Maximum Hertzian pressure, p_h	1.05G Pa	3.3G Pa
Viscosity index, α	$1.7 \times 10^{-8}\mathrm{Pa}^{-1}$	$1.7 \times 10^{-8}\mathrm{Pa}^{-1}$
Pressure-viscosity index, z	0.69	0.69

Table 4.2: Validation of line contact results. $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$

Venner [102]				This model				
n	H_c	H_m	P_s	n_p	Total dof	H_c	H_m	P_s
1793	.	0.07404	0.758	1792	308435	0.08427	0.07361	0.78420
3585	.	0.07385	0.787	3584	552191	0.08428	0.07364	0.86492
7169	.	0.07375	0.825	7168	904665	0.08428	0.07365	0.87872
14337	.	0.07370	0.850	14336	1314207	0.08428	0.07365	0.88551
28673	.	0.07367	0.867	28674	1628545	0.08428	0.07365	0.89533

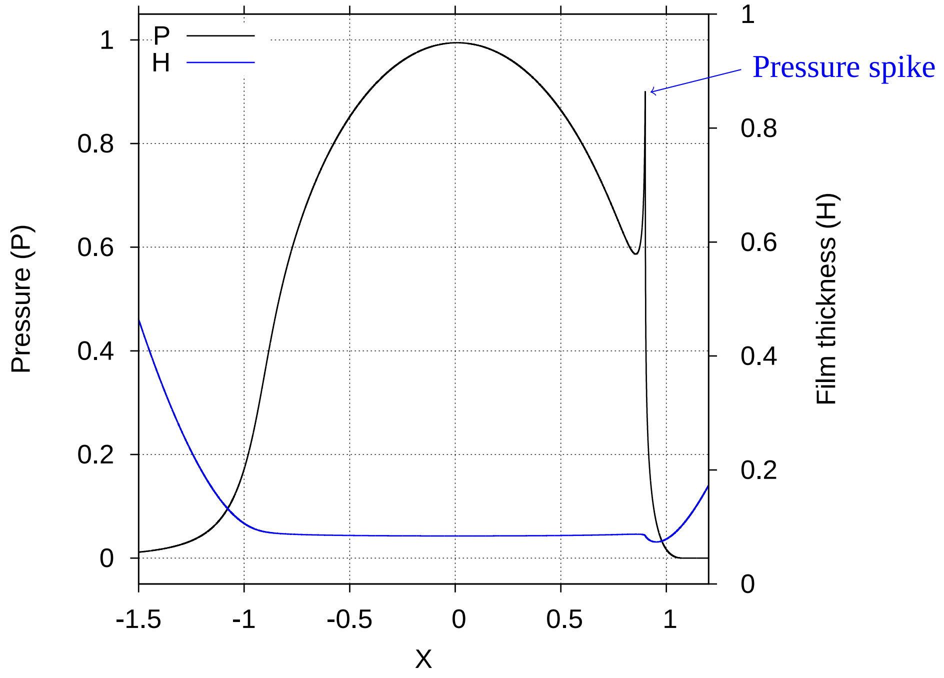


Figure 4.2: Pressure and film thickness profiles: $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$

for the Test Case 1 which is comparatively a moderately loaded case. The mesh size (with n_p finite element pressure unknowns) is kept constant throughout the whole of the fluid domain Ω_f: $X = [-4.0 : 1.5]$, in order to have a fair comparison against the finite difference based model. Table 4.2 gives a comparison of results from both models in terms of the central film thickness (H_c), the minimum film thickness (H_m) and the height of the pressure spike (P_s) (see Figure 4.2).

Venner [102] showed that the minimum film thickness is converging to 0.07365 with

Table 4.3: Validation of line contact results. $M = 200$, $L = 10$ and $p_h = 3.3\,$G Pa

Venner [103]			This model				
n	H_c	H_m	n_p	Total dof	H_c	H_m	P_s
1791	5.979×10^{-3}	5.466×10^{-3}	1791	282664	6.000×10^{-3}	5.448×10^{-3}	0.3167
3583	5.998×10^{-3}	5.502×10^{-3}	3584	426173	6.010×10^{-3}	5.480×10^{-3}	0.3073
7167	6.003×10^{-3}	5.512×10^{-3}	7167	700542	6.013×10^{-3}	5.491×10^{-3}	0.3579
14335	6.004×10^{-3}	5.516×10^{-3}	14335	1214206	6.014×10^{-3}	5.495×10^{-3}	0.3656
28671	6.005×10^{-3}	5.517×10^{-3}	28671	1628542	6.014×10^{-3}	5.497×10^{-3}	0.3931

$n = 114689$ (the number of finite difference unknowns for the pressure) and the corresponding height of the pressure spike is 0.879. The current model's results are close to these showing the validation and convergence of the current solver. A small difference in the calculated results can be expected due to use of two different discretization methods and the use of different elastic models. For the finest resolution case reported in Table 4.2 the relative difference between the computed minimum film thickness solutions is about 0.03% while that in the pressure spikes is about 3%. In a different experiment reported in [103], Venner et al. showed that the converged value of central film thickness is 0.08401 with $n = 28673$ which only differs by 0.3% with the solution computed using the current model.

As a next test we consider a more heavy loaded case (Test Case 2 given in Table 4.1) with the fluid domain Ω_f: $X = [-2.5 : 1.5]$. The mesh size (corresponding to different resolution cases) is again kept constant throughout the fluid domain Ω_f while a sufficiently fine mesh is used in the elasticity domain. A comparison of results of current model against a finite difference based model [103] is given in Table 4.3 in terms of the central film thickness (H_c) and the minimum film thickness (H_m). As for the Test Case 1 the computed results are again very close to those of published results, and that they again appear to converge to same solution. The total relative difference between the converged values is about 0.15% for the central film thickness (H_c) and 0.36% for the minimum film thickness (H_m).

Finally note that the size of the mesh in the 2D elasticity domain may be shown to be "sufficiently fine": i.e. a further decrease in the mesh size within the elasticity domain does not lead to a significant improvement in the accuracy (relative to the discretization error). In fact it is even possible to obtain the solution at low cost using much coarser meshes (like the one shown in Figure 4.1) without compromising on the accuracy of the solution (this is discussed in more detail in the next chapter).

Table 4.4: Non-dimensional parameters for the contact between steel surfaces [104].

Parameters	Test Case 3	Test Case 4
Moes parameter, L	10	10
Moes parameter, M	20	200
Maximum Hertzian pressure, p_h	0.45GPa	0.97GPa
Viscosity index, α	2.2×10^{-8}Pa^{-1}	2.2×10^{-8}Pa^{-1}
Viscosity at ambient pressure, η_0	0.04 Pa s	0.04 Pa s
Total speed, u_s	1.6 m s^{-1}	1.6 m s^{-1}

Table 4.5: Validation of point contact results: Test Case 3.

Venner [104]			This model			
$n_x \times n_y$	H_c	H_m	n_p	Total dof	H_c	H_m
64×64 (=4096)	0.41904	0.28622	4450	10486550	0.42306	0.28921
128×128 (=16384)	0.42872	0.29094	17732	14834838	0.42999	0.29123
256×256 (=65536)	0.43116	0.29218	67350	47440138	0.43129	0.29202

4.6.2 Point Contact

In this subsection, a comparison of the fully-coupled finite element point contact solver is made with published results using the integral (half-space) approach in a finite difference based model [104]. This will enable us to validate the calculated point contact results. The meshes used for this purpose have been generated with a uniform mesh size (corresponding to three different resolutions) everywhere in the contact region ($\Omega_f = [-4.5, 1.5] \times [-3, 3]$), which therefore provides a fair comparison with the finite difference based model [104]. A comparison of the obtained results with the finite difference based model [104], is given in Table 4.5 in terms of central and minimum film thicknesses (this is the only data provided in [104]) for the Test Case 3 (given in Table 4.4). It should be noted that the number marked 'n_p' represents the number of pressure unknowns in the contact region and 'Total dof' represents total unknowns of the fully coupled system. It can be seen that the calculated results are very close to that of the finite difference based model. The difference between the two solutions decreases as we switch to higher resolutions. For example, in the finest resolution case the relative difference between the central film thickness H_c solutions and the minimum film thickness H_m solutions is about 0.03% and 0.05% respectively. Moreover, it should be noted that both models use totally different approaches, therefore a small difference in the solutions is expected even though they appear to converge to the same result.

A similar trend in the solution is observed for Test Case 4 which is a comparatively more heavily loaded case. Table 4.6 provides a comparison of results obtained on fine meshes against the finite difference based model [104]. It is evident that the calculated values are again close to those of finite difference based model and that they again appear to converge to the same result.

Table 4.6: Validation of point contact results: Test Case 4.

Venner [104]			This model			
$n_x \times n_y$	H_c	H_m	n_p	Total dof	H_c	H_m
64×64 (=4096)	0.070686	0.033080	4450	10486550	0.070251	0.028999
128×128 (=16384)	0.078872	0.037120	17732	14834838	0.078987	0.036286
256×256 (=65536)	0.080935	0.038480	67350	47440138	0.081202	0.038426

Again as for the line contact case, the main drawback of these results is the use of very fine meshes, leading to very large discrete problems, which are very expensive from a computational point of view. This issue is explicitly addressed in Chapter 6.

4.7 Conclusion

In this chapter, we have discussed a standard finite element discretization of the EHL line and point contact equations. A Streamline Upwind Petrov-Galerkin (SUPG) method [22] is explained in order to stabilize the discretization of the Reynolds equation. The full system approach results in a large coupled nonlinear system (4.21) to which a Newton procedure is applied for the solution. For the iterative solution of the linearized systems (4.22) at each Newton iteration we have proposed a new preconditioner $\mathbf{P}$ in order to efficiently solve the preconditioned system (4.24) which will be demonstrated in the subsequent chapters. We have described an overall layout of the nonlinear solver along with the different variants that have been considered in this work. Finally, the accuracy of our computed results is justified by comparing them with previously published results in literature. We have shown that the difference in both our computed and the published results is small and that they appear to converge to the same solution as finer meshes are used. The only drawback of the computed results is the use of very fine meshes in the corresponding elasticity domains which is further discussed in the next chapters.

Chapter 5

Line Contact Problems

5.1 Introduction

Recall from previous chapter the development of three variants of the Newton solver where the only differences are in the attachment of inner linear solvers (direct and iterative) and the type of preconditioning of the elasticity block (AMG or GMG). The stopping tolerance for the Newton iteration is chosen as the default, i.e. the Newton iteration will successfully terminate when the maximum norm of the residual vector is lower than $U^{\frac{1}{3}}$ where U is machine unit roundoff [57]. Unless stated otherwise, for iterative variants of the solver the stopping tolerance for the GMRES iteration is $(\eta_n + U)\|F\|$. Since the preconditioning is always undertaken to the right, this stopping criterion is independent of the preconditioner used.

Note that the convergence of the GMRES method depends upon a sufficiently large dimension of Krylov-subspace being used [94]. Furthermore, at each GMRES iteration, an orthogonalization of a search direction is required against all the previous search directions, which grows the computational cost at each iteration. In other words an increase in the maximum dimension of the Krylov-subspace increases both the memory usage and the computational cost. Hence, the method becomes impractical whenever the maximum dimension of the Krylov-subspace used is large. This situation is often avoided by the use of restarted GMRES where an upper bound m is specified for the dimension of Krylov-subspace used, and if the convergence is not achieved for $k \leq m$ then GMRES is

Table 5.1: Non-dimensional parameters for different loaded line contact cases [102, 103].

Parameters	Test Case 1	Test Case 2
Moes parameter, L	10	10
Moes parameter, M	20	200
Maximum Hertzian pressure, p_h	1.05G Pa	3.3G Pa
Viscosity index, α	1.7×10^{-8}Pa^{-1}	1.7×10^{-8}Pa^{-1}
Pressure-viscosity index, z	0.69	0.69

restarted with new initial guess x_0 set to x_m. This process is repeated until convergence is achieved. One way other than the restarted GMRES may be that the maximum dimension of the Krylov-subspace should be fixed to a suitable constant, and the Newton update is applied even if the GMRES iteration is not fully converged. Note that alternative Krylov subspace methods, such as Bi-CGSTAB [101], may be used to overcome the memory issue (see Section 5.5).

For the results presented in this chapter, iterative line contact solvers are applied with a sufficiently large maximum Krylov dimension to show the effectiveness of the preconditioning strategy by ensuring that we get full convergence to the solution of the linear system (4.22) at each Newton step. We will see that the cumulative number of linear iterations appears to be independent of problem size. It should also be noted that if the initial guess is not sufficiently accurate then some under-relaxation of the outer Newton iteration may be required to achieve convergence. In this work, we used a Hertzian pressure profile (see chapter 2) as an initial guess for pressure and no under-relaxation was required to reach the converged solutions for any of the cases reported in this chapter.

An example of a typical finite element mesh used in this work is shown in Figure 5.1 which shows a fine mesh close to the contact region and a relatively coarse mesh elsewhere. The test cases considered in this chapter are taken from [102, 103], and are described in Table 5.1 in terms of Moes parameters M and L and a maximum Hertzian pressure p_h. Note that a Poisson ratio 0.3 has been used (if not explicitly stated otherwise) and the Young's modulus is then obtained according to equation (2.39) for the non-dimensional equivalent elasticity problem. Moreover, the penalty parameter $\xi = 10^6$ is used in this work. An increase in this factor will lead to an increase in the computational work for minimal change in the solution. Finally, while updating the density and viscosity, the very small negative pressures in the cavitation regions are treated as zero.
In the following section the solution of typical EHL line contact problems is discussed. This includes a discussion on the accuracy of the line contact results using coarse meshes, along with the effect of local refinement in the Hertzian contact region. It is then followed by a comparison of performances of the proposed preconditioned iterative solver compared to a state-of-the-art sparse direct solver [28] by giving a detailed comparison of

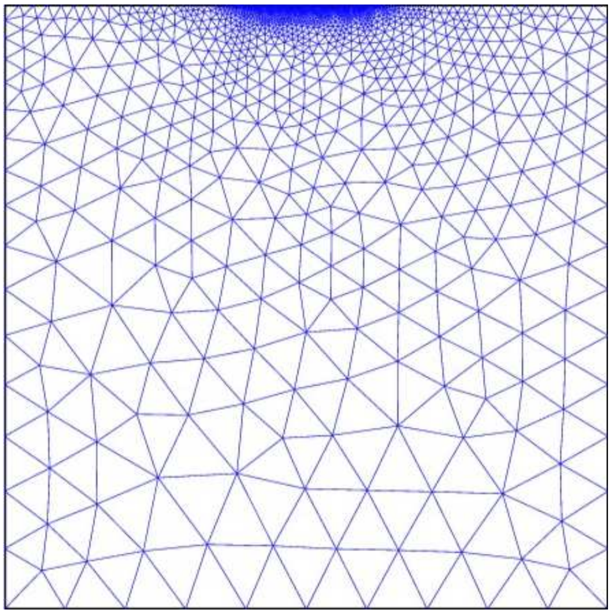


Figure 5.1: An example employed mesh for a line contact problem.

their computational times and memory usages.

5.2 Accuracy on Selected Coarse Meshes

In the previous chapter the accuracy of the line contact results for the test cases given in Table 5.1 was demonstrated by showing that the results were almost identical to previously published ones using integral approach in a finite difference based model [102,103]. However the drawback of those results were the use of very fine meshes in the elasticity domain. In fact it is possible to use much coarser meshes in the elasticity domain without compromising the accuracy of the solution. To demonstrate this the numerical solution is computed for Test Case 1 keeping the same resolution in the contact region but using large mesh sizes in the elasticity domain compared to the fine mesh cases. A comparison of these newly computed results with those using fine meshes is given in Table 5.2 in terms of central film thickness H_c, minimum film thickness H_m and the peak value of the pressure spike P_s. It is apparent from Table 5.2 that there is a minimal change in the newly computed solutions with a large decrease in the size of the elasticity problem corresponding to different resolution cases. In the finest resolution case the relative differences between the two solutions are 0.02%, 0.005% and 0.6% for H_c, H_m and P_s respectively, with a total reduction in the size of problem being almost a million degrees of freedom.

So far the mesh size was kept constant throughout the fluid domain Ω_f in order to obtain a comparison with the finite difference based model [102, 103]. As a next step,

Table 5.2: Accuracy of the line contact results over the use of fine and coarse meshes: $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$

	Fine Meshes				Coarse Meshes			
n_p	Total dof	H_c	H_m	P_s	Total dof	H_c	H_m	P_s
1792	308435	0.08427	0.07361	0.78420	45221	0.08427	0.07362	0.7972
3584	552191	0.08428	0.07364	0.86492	84491	0.08428	0.07366	0.8692
7168	904665	0.08428	0.07365	0.87872	207533	0.08428	0.07365	0.8767
14336	1314207	0.08428	0.07365	0.88551	314303	0.08427	0.07366	0.8940
28674	1628545	0.08428	0.07365	0.89533	756109	0.08426	0.07365	0.9010

Table 5.3: Line contact results with one level of refinement in the contact region [-1:1], $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$.

Uniform refinement					With local refinement				
n_p	Total dof	H_c	H_m	P_s	n_p	Total dof	H_c	H_m	P_s
1792	45221	0.08427	0.07362	0.7972	1412	36541	0.08429	0.07365	0.8043
3584	84491	0.08428	0.07366	0.8692	2825	65534	0.08429	0.07367	0.8693
7168	207533	0.08428	0.07365	0.8767	5648	125375	0.08427	0.07365	0.8792
14336	314303	0.08427	0.07366	0.8940	11296	241401	0.08426	0.07365	0.8949

we study the effect of local refinement in the non-dimensional Hertzian contact region on the accuracy of the EHL solution. This involves one level of refinement in the contact region $[-1 : 1]$ (i.e the mesh size is halved) and the accuracy of results is compared with the results given in Table 5.2. The rationale behind this is to get a significant decrease in the number of unknowns in the 1D and 2D domains without any change in the accuracy of solution. In other words this involves placing (relatively) more points only in the region where the solution has largest variation. Table 5.3 gives a comparison of results (obtained using both the uniform refinement and the non-uniform refinement in the fluid domain) in terms of the central film thickness (H_c), the minimum film thickness (H_m) and the maximum height of the pressure spike (P_s). One can see that results using the local refinement are very close to those obtained using a uniform refinement in the whole fluid domain Ω_f. Thus the local refinement leads to a significant further decrease in the problem size without significantly affecting the accuracy of the solution.

So far we discussed the accuracy of the computed EHL solution for a moderately loaded Test Case 1 using suitably coarser meshes. As a next step, we compare the accuracy of solutions for Test Case 2 (given in Table 5.1) using coarse 2D meshes against the solution obtained on fine meshes (see previous chapter). Note that the meshes used for Test Case 2 are different than those used for Test Case 1 due to considering two different fluid domains. Table 5.4 gives a comparison of results computed over both fine and coarse meshes in terms of the central film thickness (H_c), the minimum film thickness (H_m) and the maximum height of the pressure spike (P_s). Again this can be seen that a significant decrease in the size of problem for each resolution case does not lead to a significant drop

Table 5.4: Accuracy of the line contact results over the use of fine and coarse meshes: $M = 200$, $L = 10$ and $p_h = 3.3\,\text{G Pa}$

	Fine mesh solution				Coarse mesh solution			
n_p	Total dof	H_c	H_m	P_s	Total dof	H_c	H_m	P_s
1791	282664	0.006000	0.005448	0.3167	35826	0.006008	0.005456	0.3055
3584	426173	0.006010	0.005480	0.3073	68095	0.006014	0.005485	0.3405
7167	700542	0.006013	0.005491	0.3579	193752	0.006015	0.005494	0.3699
14335	1214206	0.006014	0.005495	0.3656	340932	0.006017	0.005498	0.3867
28671	1628542	0.006014	0.005497	0.3931	622378	0.006013	0.005497	0.4025

in the accuracy of EHL solution (except the height of the pressure spike). For the finest resolution case reported here, the relative change in the solution is about 0.02%, 0% (to 6 decimal places) and 2.3% in the central film thickness, the minimum film thickness and the spike height respectively, with a total reduction of about one million degrees of freedom in the size of problem. Note that the use of much coarser meshes causes notable changes in the heights of the pressure spike, and the use of slightly less coarse meshes may help to decrease the said differences.

5.3 Performance of Solvers

In this section, we discuss the performance of our preconditioned iterative solvers compared to a sparse direct solver. For each of the test cases reported in this chapter, the only initial guess we used is the Hertzian pressure profile for pressure solution and a small positive value for H_0. For the preconditioned iterative solvers, the forcing term $\eta_n = 10^{-6}$ is used for the Test Case 1 to obtain the converged solutions corresponding to various resolution cases. However, for the Test Case 2, which is a more heavily loaded case with the Hertzian pressure of about $3.3\,\text{G Pa}$, a further decrease in this parameter is required to determine a good Newton step (especially during the first few Newton iterations) to reach convergence. The reason for this is that the quality of the initial guess for which the initial residual norm $\|F\|$ is about $\mathcal{O}(10^6)$ therefore for the stopping tolerance $((\eta_n + U)\|F\|)$ of linear solver the value of η_n needs to be chosen sufficiently small (about $\mathcal{O}(10^{-9})$) such that a good Newton step is determined to avoid the risk of divergence of the Newton iteration. Practically such a small value incurs several drawbacks. These drawbacks include oversolving for the Newton step (as the Newton iteration progresses) and possibly that the linear solver may stall, especially in the final few Newton iterations due to roundoff errors of double precision arithmetic. As stated in the previous chapter, in order to cure such negative aspects one possibility may be to keep the maximum dimension of the Krylov subspace to be fixed at a suitable constant which defines an upper bound for the computa-

Table 5.5: Comparison of sparse direct and iterative (AMG) results: $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

n_p	Total dof	Sparse Direct Solver				Prec. Iterative Solver (AMG)			
		ni	t (s)	t/ni	mem.(Mb)	ni-li	t (s)	t/ni	mem.(Mb)
1792	45221	12	7.35	0.61	67	12-254	8.43	0.70	52
3584	84491	12	14.13	1.18	129	12-238	15.13	1.26	93
7168	207533	12	39.31	3.28	287	12-214	35.76	2.98	227
14336	314303	12	61.15	5.10	411	12-195	50.11	4.18	353
28674	756109	12	177.84	14.82	1028	12-191	128.07	10.67	802
28674	1628545	12	562.91	46.91	2505	12-189	279.99	23.33	1709

tional work per nonlinear iteration. Another possibility which is used in this experiment (for Test Case 2) is to fix the stopping tolerance of linear solver $(\eta_n + U)\|F\| = 10^{-4}$ until the residual norm $\|F\|$ do not become sufficiently small. In other words the inexact Newton strategy is implemented in the final few Newton iterations where the value of η_n becomes constant and the drop in the tolerance of linear solver depends only on $\|F\|$. Our experience shows that this is quite advantageous to avoid oversolving of the linear systems at each Newton iteration.

Finally, the comparison of different variants of the solver is split into two phases which are described in detail in the following subsections for each of the test cases reported in Table 5.1. Note that all timings reported here were computed using an Intel Xeon CPU W3520 @ 2.67GHz with 6 GB RAM. This is a different architecture to that used to obtain the timings reported in [2].

5.3.1 Algebraic Multigrid Solver

In the first phase, a comparison of sparse direct and AMG preconditioned iterative results for Test Case 1 are given in Table 5.5 in terms of total iterations, total time, time per nonlinear iteration (ni) and memory usage. The number (li) presents the sums of linear iteration counts across all Newton steps. It can be seen that the sparse direct solver is slightly faster in the first two cases, but as the size of the problem is increased its performance deteriorates (i.e. the time and memory growth are superlinear) and it becomes less efficient than the preconditioned AMG solver. For the preconditioned AMG solver both the memory usage and the computational time grow almost linearly with increasing problem size. Furthermore, the number of linear iterations (li) are independent of the size of the problem.

As a next step we give a comparison of direct and preconditioned AMG iterative results for Test Case 2 in Table 5.6 in terms of total iterations, total time, time per nonlinear iteration (ni) and memory usage. For this heavy loaded case the computational work

Table 5.6: Comparison of sparse direct and iterative (AMG) results: $M = 200$, $L = 10$ and $p_h = 3.3\,\text{G Pa}$

n_p	Total dof	Sparse Direct Solver				Prec. Iterative Solver (AMG)			
		ni	t (s)	t/ni	mem.(Mb)	ni-li	t (s)	t/ni	mem.(Mb)
1791	35826	18	8.38	0.47	56	20-538	13.40	0.67	43
3584	68095	19	17.22	0.91	103	18-445	21.26	1.18	77
7167	193752	18	57.84	3.21	277	18-428	60.61	3.37	221
14335	340932	18	105.03	5.84	459	19-453	113.43	5.97	403
28671	622378	18	234.10	13.01	864	19-443	209.46	11.02	703
28671	1628542	22	1230.89	55.95	2676	23-452	611.05	26.57	1799

per nonlinear iteration has increased, as expected. As for the sparse direct solution the preconditioned AMG solver also requires more work to achieve the convergence. It can be seen that the sparse direct solver is slightly more efficient (in terms of computational times) than the preconditioned AMG solver in the first four mesh resolutions, but as the problem size further increases the performance of the sparse direct solver deteriorates (as observed for the Test Case 1) and the preconditioned AMG iterative solver eventually becomes more efficient. The cumulative number of linear iterations (li) are again independent of the size of the problem. Finally, in the preconditioned AMG results both the computational time and the memory growth appear to be linear with increasing problem size.

5.3.2 Geometric Multigrid Solver

In the second phase of performance assessment we give a comparison of using AMG versus GMG in the preconditioning of the elasticity block (J_{22} in (4.23)). Starting with a suitable coarse grid a sequence of geometrically nested fine grids are obtained through uniform mesh refinement. Moreover, on each fine level, mesh files are written in order to use them to get preconditioned iterative AMG results for comparison, where AMG performs its own coarsening which results in a greater number of levels than in the GMG case. The coarsest grid uses one extra level of refinement in the Hertzian contact region in order to see the performance over unstructured meshes. Again this comparison is performed for the test cases given in Table 5.1. For Test Case 1, the results obtained from the two different preconditioners are given in Table 5.7 in terms of total iterations, total time, time per nonlinear iteration and memory usage. It should be noted that in the first case GMG preconditioning uses only 2-levels and therefore is not as efficient as AMG preconditioning. However, as the number of levels increases one can see that it performs better than the AMG preconditioning. Both the time and the memory performance are superior except for the first case. As for AMG preconditioning, both the time and memory growth

Table 5.7: Comparison of preconditioned iterative results (with AMG and GMG): $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

n_p	Total dof	Prec. Iterative Solver (AMG)				Prec. Iterative Solver (GMG)				
		ni-li	t (s)	t/ni	mem.(Mb)	ni-li	t (s)	t/ni	mem.(Mb)	levels
1417	25730	12-235	4.20	0.35	29	12-228	9.35	0.78	27	2
2835	96196	12-221	16.42	1.37	106	12-220	16.72	1.39	81	3
5671	371336	12-195	61.46	5.12	418	12-212	50.57	4.21	305	4
11343	1458448	12-187	243.48	20.29	1625	13-211	204.00	15.69	1123	5

Table 5.8: Comparison of preconditioned iterative results (with AMG and GMG): $M = 200$, $L = 10$ and $p_h = 3.3\,$G Pa

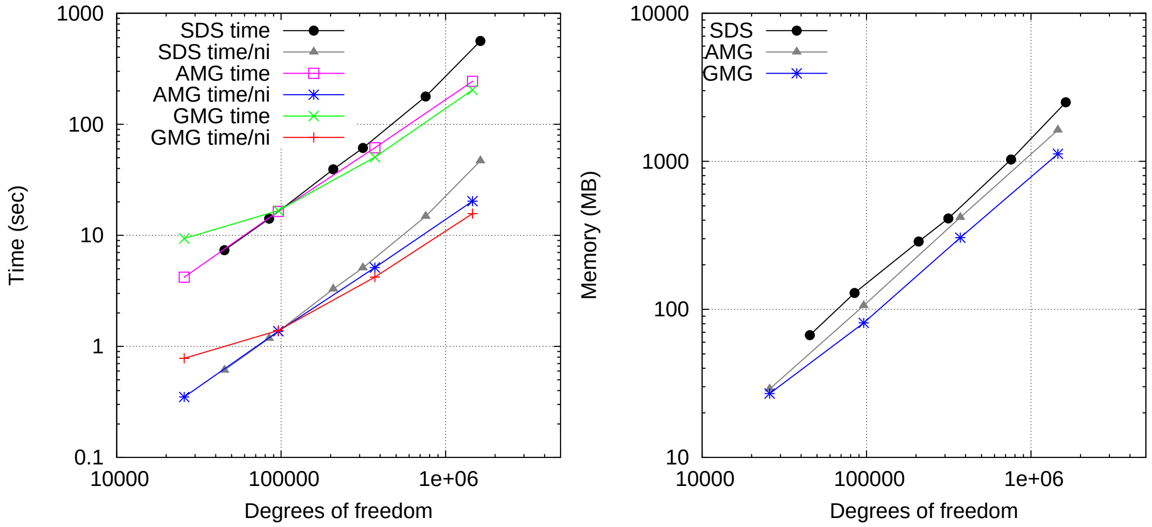
n_p	Total dof	Prec. Iterative Solver (AMG)				Prec. Iterative Solver (GMG)				
		ni-li	t (s)	t/ni	mem.(Mb)	ni-li	t (s)	t/ni	mem.(Mb)	levels
1495	23602	18-514	7.60	0.42	29	18-583	19.25	1.07	29	2
2991	87724	20-482	29.70	1.49	100	20-599	34.62	1.73	86	3
5983	337528	21-465	119.06	5.67	402	23-606	107.49	4.67	321	4
11967	1323376	21-420	452.73	21.56	1524	20-539	379.91	19.00	1272	5

is almost linear and the number of linear iterations (li) are independent of the problem size.

As for Test Case 1, a similar behaviour in the performance of the different preconditioned iterative solvers is observed for the Test Case 2. The computational details for both solvers are provided in Table 5.8. For this heavy loaded case, the preconditioned GMG solver again competes favourably with the performance of the AMG solver as the number of its levels goes up. Both the computational time and the memory usage appear to be growing linearly with the size of problem for both solvers. Furthermore, the number of linear iterations (li) are again independent of the problem size.

5.3.3 Further Discussion

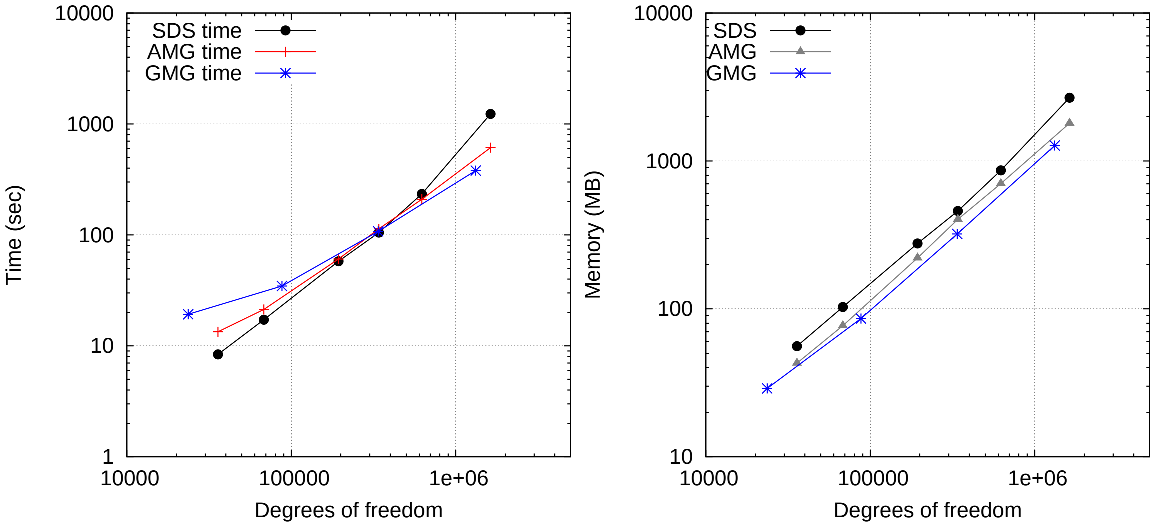
A graphical representation of the performances of the different solvers is plotted in Figure 5.2(a) for the Test Case 1, showing total time and time per nonlinear iteration for the different solvers. It can be seen that the performance of the direct solver deteriorates most quickly with increasing problem size. Moreover, the performance of the GMG solver is poor on coarse meshes but as the number of levels goes up it performs well compared to the other solvers. Similarly, the observation from Figure 5.2(b) reveals that both iterative solvers are more memory efficient in all mesh cases than the sparse direct solver, and the GMG solver requires even less memory than the AMG solver. Finally, the efficiency of each preconditioned iterative solver is quite close to optimal as both the memory and time growth appear to be linear with increasing problem size (ignoring the GMG cases with small numbers of levels).



(a) Computational time

(b) Memory usage (MB)

Figure 5.2: Performances of different line contact solvers: $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa



(a) Computational time

(b) Memory usage (MB)

Figure 5.3: Performances of different line contact solvers: $M = 200$, $L = 10$ and $p_h = 3.3\,$G Pa

A similar observation can be made for the Test Case 2, for which a graphical representation of the computational time (total time) and memory usage of the different solvers is given in Figure 5.3(a) and Figure 5.3(b) respectively. Again the sparse direct solver is seen to be efficient for smaller problems but its efficiency deteriorates as the problem size increases. Thus the iterative solvers become more efficient than the sparse direct solver. Furthermore, the GMG solver performs even better than the AMG solver once it exceeds a certain number of grid levels. Finally, even for this heavy loaded case, both the time and the memory growth again appear to be linear with the increasing problem size, justifying the performance of iterative solvers as being close to the optimum level.

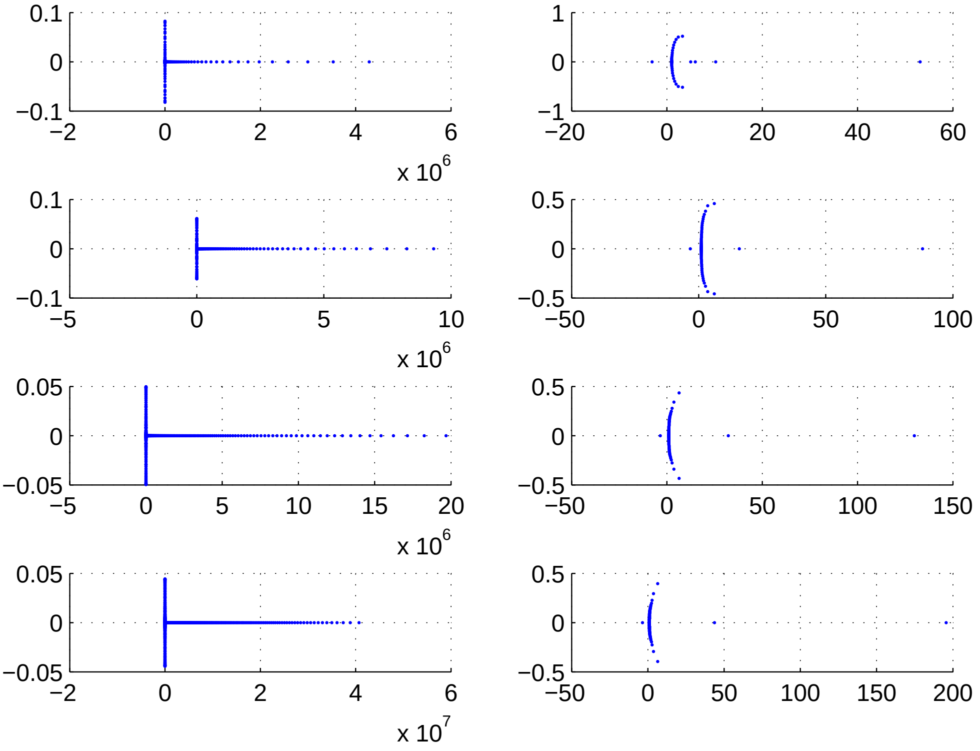
Note however that the GMG approach is not quite as effective as it may appear from an initial inspection of Table 5.7 and Figure 5.2(a), for example. The need to generate a nested sequence of grids puts unnecessary restrictions on the finest mesh and so many more degrees of freedom are required for a given accuracy. For example, the finest case in Table 5.7 has $n_p = 11343$ but $n_u = 723552$. With an unstructured mesh that has more unknowns in Ω_f ($n_p = 14336$ in Table 5.5) only 314303 degrees of freedom are needed in total (i.e. $n_u = 149983$). In this specific example, the AMG solver with least problem size leads to about 75% savings in both the CPU and the memory usage compared to the GMG solver with a large problem size.

5.3.4 Eigenvalues Analysis

So far, it has been shown that the performance of preconditioned iterative solvers appeared to be nearly optimal. This showed the effectiveness of the preconditioning strategy used in this work. Recall that a good preconditioner has an effect of clustering the eigenvalues of a system matrix within a small interval (or a small number of small intervals). In this subsection, the behaviour of the eigenvalues of both the original and the transformed (preconditioned) system is discussed as the discretization goes finer. Note that a total of four different discretizations are used for this purpose.

Figure 5.4 shows the distribution of eigenvalues (for the starting Newton iteration) of the original (left) and the preconditioned system (right) as the discretization becomes finer (top to bottom). It can be seen that the eigenvalues of the original system are well distributed in a large interval of size about $\mathcal{O}(10^6)$. Moreover, the magnitude of the largest eigenvalue roughly doubles as the discretization becomes finer which means that the original system is getting more and more ill-conditioned. On the other hand, one can see that the eigenvalues of the preconditioned system are well clustered within a small number of small intervals. Note that the majority of the eigenvalues are clustered around 1 compared to the original system and that the magnitude of the largest eigenvalue does not grow large. In other words, the preconditioning operator appears to have a good effect on clustering the eigenvalues and that is why such an optimal performance is experienced with the proposed preconditioner.

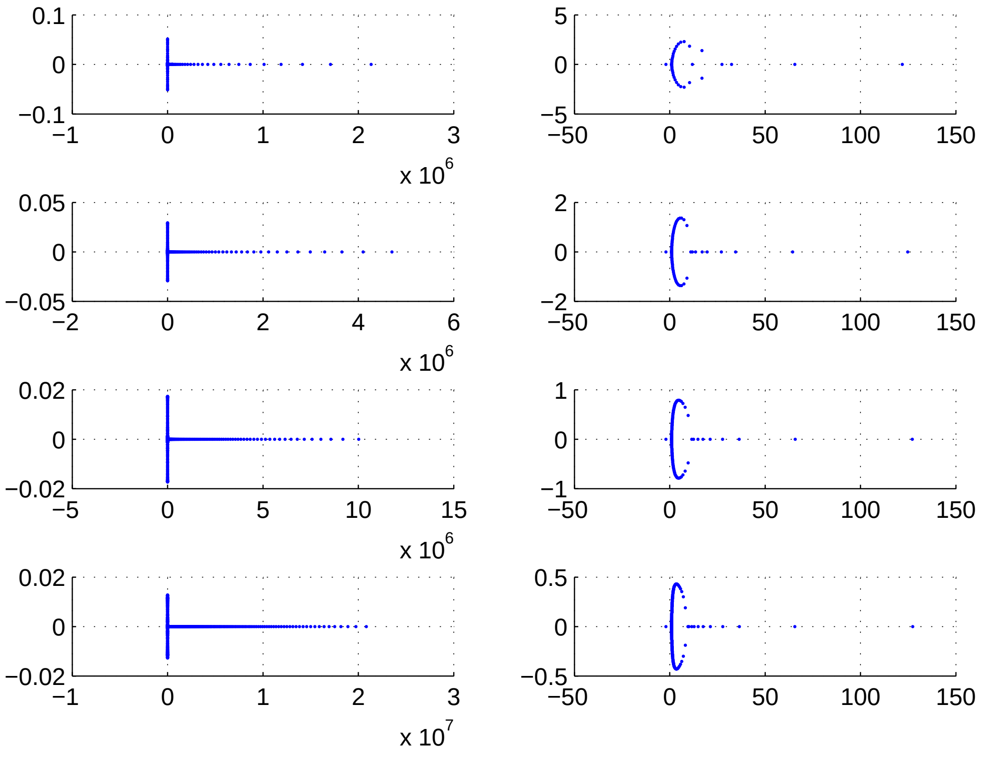
As a next case, Figure 5.5 shows the distribution of eigenvalues of both the original and preconditioned system at one of the last Newton iterations. The same behaviour in



(a) Original system (b) Preconditioned system

Figure 5.4: The behaviour of eigenvalues of the original and preconditioned system at the starting Newton iteration with successive finer discretizations: $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

the distribution of eigenvalues can be observed both for the original and the preconditioned system. For the original system, the magnitude of largest eigenvalue gets twice as large as the discretization goes finer. In the case of the preconditioned system, the majority of eigenvalues are again clustered close to 1, and the magnitude of the largest eigenvalue remains the same with the finer resolution. Furthermore, one can observe that the eigenvalues get more clustered close to 1 as the resolution of the problem increases. In other words, the preconditioned system at the finer discretizations appears to have a better spectrum compared to the coarse discretizations. Overall, the effectiveness of the preconditioning appears not to be affected as the Newton iterations progress.



(a) Original system (b) Preconditioned system

Figure 5.5: The behaviour of eigenvalues of the original and preconditioned system at the 10th Newton iteration with successive finer discretizations: $M = 20$, $L = 10$ and $p_h = 1.05\,\text{G Pa}$

5.4 Varying Poisson Ratio: Accuracy and Performance

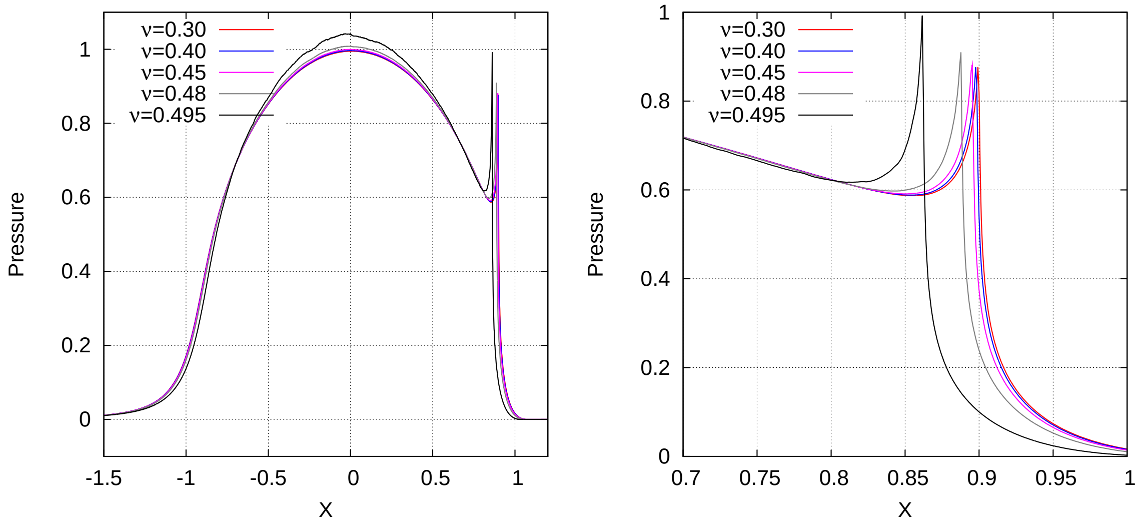
In the results presented so far, a Poisson ratio of 0.3 has been used for calculating the elastic deformation of elastic material. In this section we study the effect of varying the Poisson ratio for the elastic material on the accuracy and the performance of current model. This will be achieved by varying the Poisson ratio between 0.25 and 0.495. Recall that the material reaches the incompressibility limit as the Poisson ratio approaches 0.5.

5.4.1 Accuracy

In this subsection we discuss the effect of variation of Poisson ratio over the accuracy of EHL solution. For the sake of demonstration, Test Case 1 is run for different values of Poisson ratio for the 2D elastic material using 7168 unknowns for pressure and 207533 degrees of freedom in total $(1 + n_p + 2n_u)$. The results are presented in Table 5.9 in terms

Table 5.9: Effect of varying the Poisson ratio over the accuracy of the line contact results with $n_p = 7168$ and 207533 degrees of freedom in total: $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$

ν	H_c	H_m	P_s	H_0
0.25	0.08428	0.07366	0.8731	−2.4307
0.30	0.08428	0.07365	0.8767	−2.4204
0.35	0.08427	0.07365	0.8753	−2.4059
0.40	0.08427	0.07365	0.8761	−2.3847
0.45	0.08426	0.07364	0.8810	−2.3490
0.47	0.08424	0.07364	0.9100	−2.3219
0.48	0.08422	0.07364	0.9097	−2.2979
0.49	0.08414	0.07363	0.9246	−2.2466
0.495	0.08399	0.07357	0.9923	−2.1767



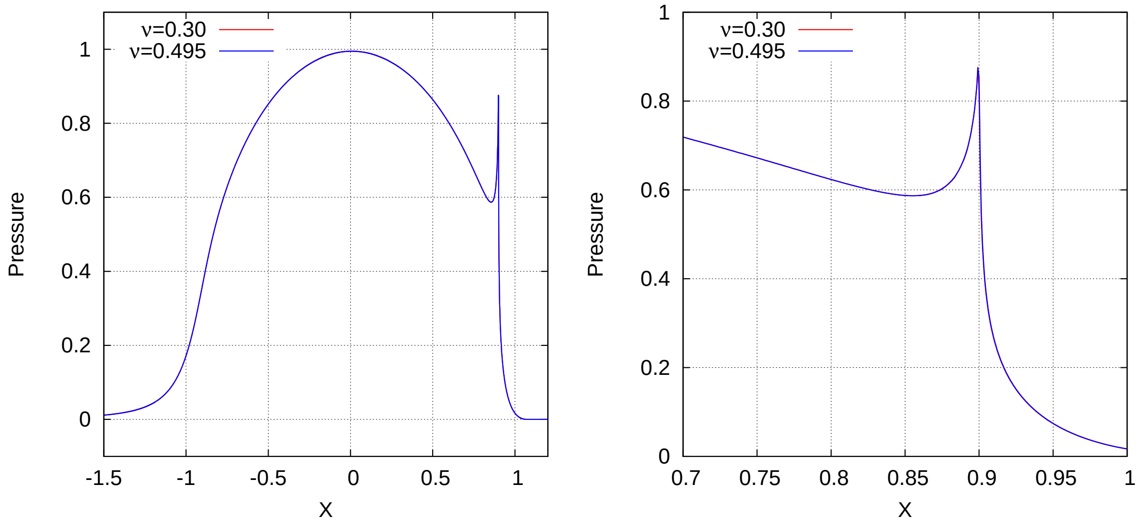
(a) Pressure profiles

(b) A close-up of pressure profiles

Figure 5.6: Pressure profiles for different values of Poisson ratio with $n_p = 7168$ and 207533 degrees of freedom in total: $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$

of central film thickness H_c, the minimum film thickness H_m and the height of pressure spike P_s. It can be noticed that for the different values of Poisson ratio between 0.25 and 0.40 there are only minor changes in the solution, but as the Poisson ratio exceeds 0.40 these changes in the solution become significant. This can especially be observed for $\nu = 0.495$ with a large jump in the height of pressure spike though that jump does not appear in the film thickness solutions (both H_c and H_m). However this can be clearly seen in Figure 5.6 which suggests that the pressure solution is not fully converged as the Poisson ratio approaches 0.5.

This kind of behaviour in the solution (for harder materials) is due to the inaccurate elastic deformation solutions within this situation. In such situations the use of a sufficiently fine mesh in the elasticity domain helps to overcome this problem. Note that such



(a) (b)

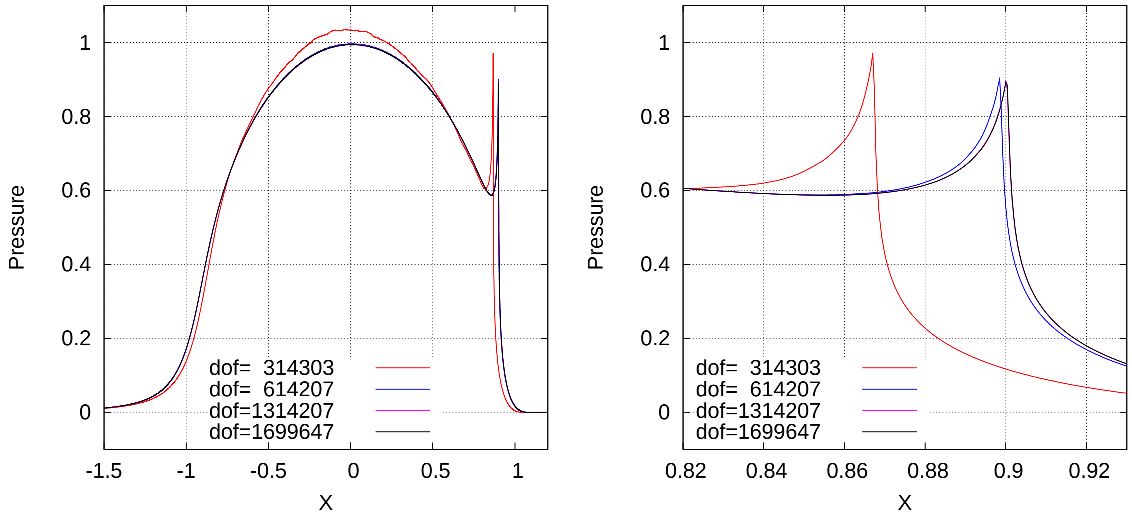
Figure 5.7: Pressure profiles for $\nu = 0.30$ and $\nu = 0.495$ computed over coarse mesh (with $n_p = 7168$ and total dof = 207533) and fine mesh ($n_p = 7168$ and total dof = 904665) respectively: $M = 20$, $L = 10$ and $p_h = 1.05$ G Pa

Table 5.10: Computed solution for $\nu = 0.30$ and $\nu = 0.495$ computed over coarse mesh (with $n_p = 7168$ and total dof = 207533) and fine mesh ($n_p = 7168$ and total dof = 904665) respectively: $M = 20$, $L = 10$ and $p_h = 1.05$ G Pa

ν	H_c	H_m	P_s	H_0	D_{max}
0.30	0.08428	0.07365	0.8767	-2.4204	2.5047
0.495	0.08426	0.07366	0.8753	-2.3383	2.4225

fine meshes are obtained by keeping a specific resolution unchanged in the contact region and using smaller mesh sizes in the remaining part of the elasticity domain. To justify this, an experiment is carried out for $\nu = 0.495$ using the same resolution (i.e. 7168 unknowns for pressure) in the fluid domain but adding more points in the elasticity domain leading to 904665 degrees of freedom in total.

The previously obtained result for $\nu = 0.30$ (with a coarse 2D mesh) is plotted together with newly computed solution for $\nu = 0.495$ (with a much finer 2D mesh) and is shown in Figure 5.7. One can see that the results shown now appear to have converged to the same solution. Moreover, a comparison between their computed values is listed in Table 5.10 where D_{max} represents the maximum elastic deformation of the equivalent contact surface. It should be noted that the elastic deformation profile of the contact surface for $\nu = 0.495$ tends to differ from that for $\nu = 0.3$ only by a small constant and this small constant is balanced by their corresponding values of H_0. A similar trend is observed in the other resolution cases where a use of very fine 2D mesh leads to matching



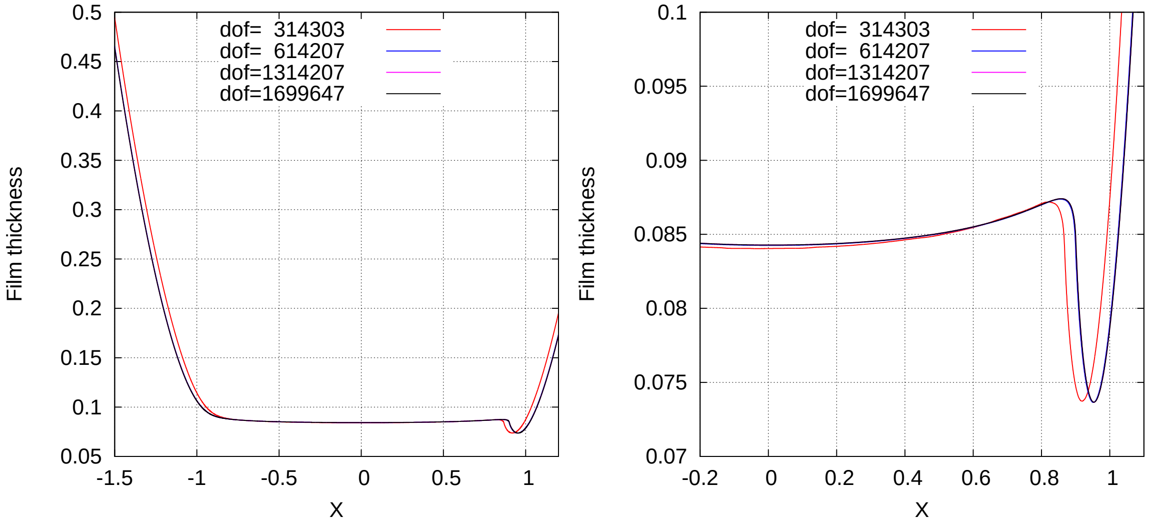
(a) (b)

Figure 5.8: Convergence of the pressure profile for $\nu = 0.495$ with increasing problem size: $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$

the EHL solution for $\nu = 0.495$ to that with $\nu = 0.3$ on a much coarser mesh. Moreover, this behaviour can also be justified by considering the non-dimensional half space method which has no explicit dependency either on the Young's modulus or the Poisson ratio but yields a fixed elastic deformation profile (assuming fixed operating pressure). Furthermore one should not be surprised with the value of maximum elastic deformation reported in Table 5.10 because the numerical solution of the elasticity problem leads to the displacement proportional to $\ln r$, see for example [33, 66] i.e the numerical solution of the elasticity problem accommodates all those constants which have been adjusted or combined with H_0 in the traditional half space approach.

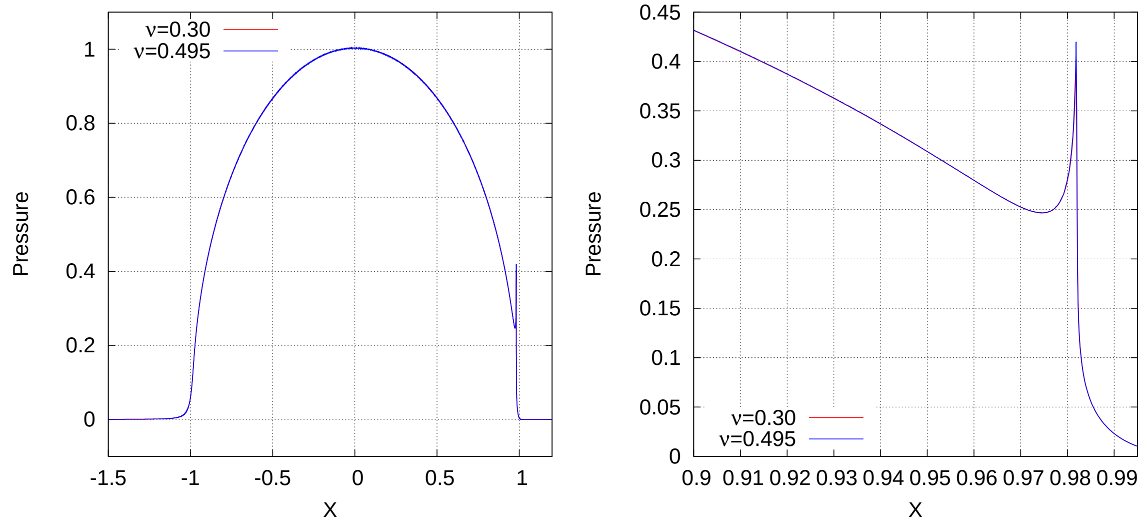
So far, it was shown that the use of very fine 2D mesh for large Poisson ratio leads to achieve the same accuracy in the EHL solution compared to that with small Poisson ratio using a coarse 2D mesh. In order to see the convergence of the EHL solution for higher values of the Poisson ratio as the elasticity mesh goes finer and finer, Test Case 1 is run using $\nu = 0.495$ over a sequence of four meshes leading to, respectively, 314303, 614207, 1314207 and 1699647 degrees of freedom in total. Note that these meshes use the same resolution in the fluid domain (14336 unknowns for pressure). The pressure profiles shown in Figure 5.8 reveals that on the coarser mesh the solution is not fully converged, but as the problem goes finer and finer the pressure profile appears to have converged to the same solution. A similar trend can also be observed in Figure 5.9 showing the convergence of film thickness profiles as the mesh goes finer and finer.

Consider Test Case 2 with the finest resolution that we have used for the pressure



(a) (b)

Figure 5.9: Convergence of the film thickness profile for $\nu = 0.495$ with increasing problem size: $M = 20$, $L = 10$ and $p_h = 1.05\,\mathrm{G\,Pa}$



(a) (b)

Figure 5.10: Pressure profiles for $\nu = 0.30$ and $\nu = 0.495$ computed over coarse mesh (with $n_p = 28671$ and total dof = 622378) and fine mesh ($n_p = 28671$ and total dof = 1628542) respectively: $M = 200$, $L = 10$ and $p_h = 3.3\,\mathrm{G\,Pa}$

solution, i.e. using 28671 unknowns for pressure. Again we consider a coarse 2D mesh yielding 622378 degrees of freedom in total and a very fine 2D mesh with 1628542 degrees of freedom in total. The solution for Test Case 2 is computed using the Poisson ratio 0.30 and 0.495 on these coarse and the fine meshes. For this heavy loaded case using $\nu = 0.495$ neither of the solvers reached convergence on the coarse 2D mesh, therefore we provide

Table 5.11: Effect of varying the Poisson ratio over the performances of different line contact solvers with $n_p = 7168$ and 904665 degrees of freedom in total: $M = 20$, $L = 10$ and $p_h = 1.05$ G Pa

n_p	Sparse Direct Solver				Prec. Iterative Solver (AMG)			
	ni	t (s)	t/ni	mem.(Mb)	ni-li	t (s)	t/ni	mem.(Mb)
0.25	12	566.64	47.22	1949	12-212	164.57	13.71	1038
0.30	12	566.64	47.22	1947	12-212	166.88	13.91	1047
0.35	12	567.58	47.30	1949	12-212	166.77	13.90	1033
0.40	12	568.36	47.36	1948	12-227	176.20	14.68	1060
0.45	12	567.19	47.27	1948	12-272	206.88	17.24	1108
0.47	12	566.70	47.23	1948	14-320	250.35	17.88	1151
0.48	12	566.84	47.24	1948	12-330	254.99	21.25	1172
0.49	12	568.55	47.38	1948	12-440	330.01	27.50	1283
0.495	12	568.81	47.40	1952	12-596	434.66	36.22	1415

a direct comparison between the solutions for $\nu = 0.30$ and $\nu = 0.495$ as computed on the coarse and the fine mesh respectively. This is shown in Figure 5.10. Again both cases appear to converge to the same solution however there is slight difference in the height of pressure spike. Moreover, despite the appearance of Figure 5.10, tiny oscillations in the solution for $\nu = 0.495$ are observed, i.e. the pressure profile is not absolutely smooth and might be causing a small difference in the height of pressure spike to that with $\nu = 0.30$ on a coarser mesh.

Overall, these experiments suggest that the use of different values of Poisson ratio for the elastic material yields the same EHL solution provided a sufficiently accurate elasticity mesh has been used.

5.4.2 Performance

In this subsection, we consider the effect of varying the Poisson ratio over the performance of the AMG preconditioned iterative solver. For this purpose Test Case 1 is considered with 7168 unknowns for pressure and 904665 degrees of freedom in total. The Poisson ratio is varied from 0.25 to 0.495. In Table 5.11, the performance of both the sparse direct and the iterative (AMG) variants of the solver are considered in terms of the total nonlinear iterations (ni), linear iterations (li) (iterative solver only), total time (t (s)), time per nonlinear iteration and the memory usage. As expected, for different values of the Poisson ratio the performance of the sparse direct solver appears to be essentially constant both in terms of the computational time and the memory usage. On the other hand the performance of the AMG solver is almost constant for the values of the Poisson ratio up to 0.40. But once the Poisson ratio exceeds this value the performance of the AMG solver deteriorates as the iterative solver requires more work to achieve the convergence at each nonlinear iteration. Since the GMRES method is used as the inner linear solver

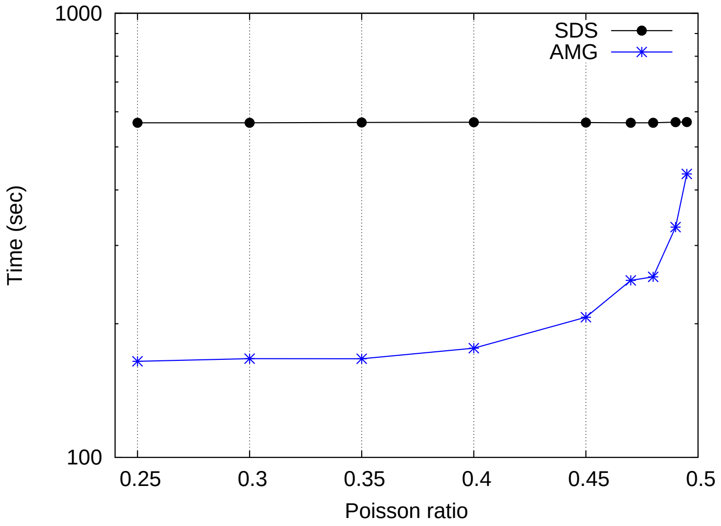


Figure 5.11: Effect of varying the Poisson ratio over the performances of different line contact solvers with $n_p = 7168$ and 904665 degrees of freedom in total: $M = 20$, $L = 10$ and $p_h = 1.05$ G Pa

this growth in the computational work also leads to a corresponding growth in the total memory usage. A more clear picture of the performance of different solvers (in terms of their computational times) is shown in the Figure 5.11. This reveals that the performance of the AMG solver deteriorates when exceeding $\nu = 0.40$ but when the Poisson ratio gets very close to 0.5 it deteriorates drastically. A similar behaviour in the performance (as for the accuracy) is also observed in the other resolution cases sketching the same story. This kind of behaviour in the performance of an AMG solver is not surprising and can be generally expected whenever the Poisson ratio is close to 0.5 [16, 48]. In other words, for higher values of the Poisson ratio the corresponding systems become more ill-conditioned [18, 48] which affects the optimality of AMG solvers.

Finally, in order to analyse the performance of the preconditioned AMG solver for a fixed value of Poisson ratio close to 0.5, an experiment is set up for a Poisson ratio $\nu = 0.48$. The results are obtained over a sequence of fine 2D meshes (corresponding to different resolutions in the fluid domain), and these are presented in Table 5.12, while a more clear picture of these results is shown in Figure 5.12. One can see that both the computational time and the memory growth again appear to be linear with increasing problem size (despite of the slightly worse CPU time for higher values of the Poisson ratio). Furthermore, Table 5.12 reveals that the cumulative number of linear iterations also appears independent of the problem size. This experiment shows that the AMG solver scales almost linearly even for the higher values of the Poisson ratio.

Table 5.12: Performance of the AMG solver for Poisson ratio $\nu = 0.48$ on a sequence of fine 2D meshes (corresponding to different resolutions in the fluid domain): $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

n_p	Total dof	ni-li	t(s)	t/ni	mem.(MB)
1792	308435	12-379	94.10	7.84	422
3584	552191	12-347	159.69	13.31	741
7168	904665	12-330	254.99	21.25	1172
14336	1699647	12-293	448.35	37.36	2167
28674	2028545	13-312	556.72	42.82	2463

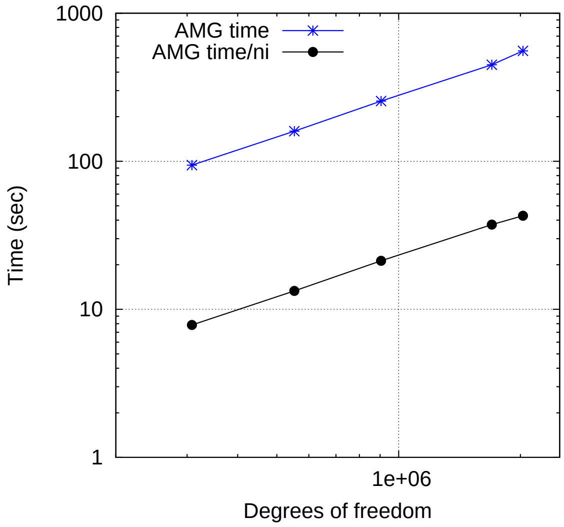


Figure 5.12: Performance of the AMG solver for Poisson ratio $\nu = 0.48$ on a sequence of fine 2D meshes (corresponding to different resolutions in the fluid domain): $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

5.5 Biconjugate Gradient Stabilized Method (Bi-CGSTAB)

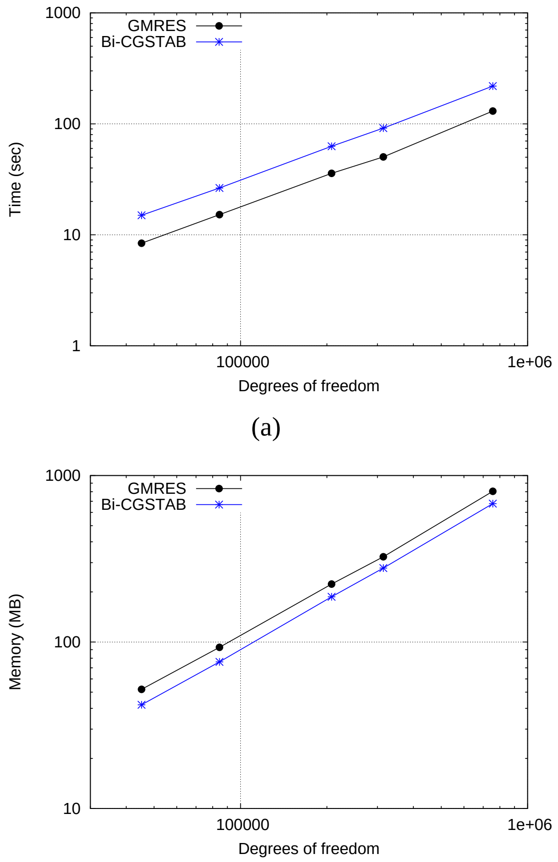
In this section the performance of different Krylov iterative solvers are discussed. Recall from Chapter 3 that the computational cost of the GMRES method grows like $\mathcal{O}(kn)$ due to orthogonalization of a vector against all the previous search directions. This situation can be avoided by using a restarted GMRES with a suitable upper bound m for the number of GMRES iterations before restarting. On the other hand the Bi-CGSTAB method does not require the storage of all previous search directions (though it still requires twice as much memory storage as compared to the conjugate gradient method for symmetric positive definite systems). Nevertheless, the method can still have advantages over the GMRES method, especially when the matter of the storage is important. Therefore, an experiment is setup for the Test Case 1 where the main linear solver uses both the GMRES and the Bi-CGSTAB as an alternative.

Table 5.13: Performance of different Krylov subspace based iterative linear solvers: $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

n_p	Total dof	GMRES				Bi-CGSTAB			
		ni-li	t (s)	t/ni	mem.(Mb)	ni-li	t (s)	t/ni	mem.(Mb)
1792	45221	12-254	8.40	0.70	52	12-283	15.01	1.25	42
3584	84491	12-238	15.20	1.27	93	12-264	26.45	2.20	76
7168	207533	12-214	35.81	2.98	223	12-241	62.79	5.23	187
14336	314303	12-195	50.34	4.20	325	12-235	91.47	7.62	278
28674	756109	12-191	130.36	10.86	804	12-220	218.67	18.22	678

The results are listed in Table 5.13 in terms of ni, li, total time, time per nonlinear iteration and the memory usage. These results are obtained using the full inexact Newton strategy with $\eta_n = 10^{-6}$ and both linear solvers are allowed to use the maximum dimension of Krylov subspace in order to give a fair comparison. The results presented in Table 5.13 are evidence of the expected behaviours of the two linear solvers in terms of further memory savings by the use of the Bi-CGSTAB method. However the method is not quite as robust as GMRES for these tests. In other words the Bi-CGSTAB method is about 75% (on average) slower than the GMRES method. Indeed the cumulative number of linear iterations across all the Newton steps are slightly more than the GMRES method and, looking at the total computational cost, also reveals that a single Bi-CGSTAB iteration is more expensive than a single GMRES iteration which may be due to twice as many as matrix-vector products and the preconditioned solves compared to a GMRES iteration.

A more clear picture about the growth in the computational cost and the memory usage is shown in Figure 5.13. Figure 5.13(a) reveals that for each resolution case the total computational cost with the use of Bi-CGSTAB method is about 75% more than that with the use of GMRES. However the growth in computational cost for both the methods appears to be linear. A similar trend is observed for the memory usage of two methods in Figure 5.13(b) where the GMRES method is seen to be slightly worse than the Bi-CGSTAB method as expected. If we compare the overall performance of the two methods then the GMRES methods appears to be far better than the Bi-CGSTAB method as one can see that a small compromise on the memory leads to significant savings in the computational times. Despite the performance, one may even face the entire failure of convergence of the Bi-CGSTAB method for heavily loaded cases, which in our case, was observed for Test Case 2. Finally, further tests would need to be undertaken to confirm that these conclusions hold across a wider range of parameter space.



(a)

(b)

Figure 5.13: Performance of different Krylov subspace based iterative linear solvers: $M = 20$, $L = 10$ and $p_h = 1.05\,$G Pa

5.6 Conclusion

In this chapter the accuracy and the performance of different line contact solvers have been discussed. First of all the accuracy of the EHL results were discussed on the coarse 2D meshes relative to finer ones which have been used in the previous chapter. We showed that the accuracy of the computed EHL solution is not significantly affected for the set of coarse meshes we used in this work. We also discussed briefly the effect of using a relatively more fine mesh in the Hertzian contact region over the accuracy and cost.

We further gave a detailed comparison of the performance of the different solvers developed for the line contact problem. It was shown that the AMG preconditioned variant of the iterative solver performs better than the sparse direct solver. We further showed that the performance significantly improves if AMG preconditioning of the elasticity block is switched to the GMG preconditioning. However, in the case of GMG preconditioning one needs to accept the restriction over the meshes when using a global refinement to generate the hierarchy of the meshes. This restriction seems to be too great to allow GMG to outperform AMG in terms of accuracy per CPU cycle.

The effect of varying the Poisson ratio (for the 2D elastic model) over the accuracy and the performance of our line contact solvers was also demonstrated. It was concluded that the variation of Poisson ratio does not affect the accuracy of EHL solution provided a sufficiently fine 2D mesh is used. Moreover, the performance of the preconditioned iterative solver deteriorates drastically whenever the Poisson ratio becomes sufficiently close to 0.5. Finally, the performance of the GMRES and Bi-CGSTAB methods was compared. It was observed that the Bi-CGSTAB method as an inner linear solver leads to a further savings in the memory usage however its performance was observed to be not so good as that of GMRES in terms of CPU time. In other words, the small compromise on the memory usage the GMRES method as an inner linear solver found to be superior for the cases considered here.

Chapter 6

Point Contact Problems

6.1 Overview

In this chapter the accuracy and performance of the solver is considered for EHL point contact problems. The issues addressed in this chapter include the selection of efficient 3D meshes based upon a series of experiments to ensure the accuracy of EHL point contact solutions at minimal computational cost. The memory and CPU savings for the proposed preconditioned Newton-Krylov approach, relative to a sparse direct Newton solver, are consistent but relatively small for EHL line contact problems. In this chapter we consider, much larger, point contact problems for which these advantages are very much more significant. Finally, we discuss the effect of the quality of the tetrahedral meshes over the accuracy of the solution.

6.2 Problem Consideration

As discussed in Chapters 4 and 5, three variants of the nonlinear solver have been developed. The first variant uses a sparse direct routine [28] as the inner linear solver while the other variants uses the GMRES [94] method for inner linear solves. The difference in the later two variants is the manner of the preconditioning of the elasticity block in the fully-coupled system, based upon AMG and GMG preconditioning respectively. Recall from Chapter 4 that the stopping tolerance for the Newton iteration is chosen as $U^{\frac{1}{3}}$ with U being the machine unit roundoff. In the iterative variants of the Newton solver the stopping tolerance for GMRES iteration is $(\eta_n + U)\|F\|$ with $\eta_n = 10^{-6}$ for the problems considered in this chapter. Moreover, as discussed previously, the convergence of the GMRES method depends upon the dimension of the Krylov-subspace used, and the method becomes more expensive whenever the maximum dimension of the Krylov-subspace used is large. For the line contact problems in the previous chapter the GMRES iteration was, however, allowed to use a sufficiently large dimension so as to achieve convergence. The purpose behind that choice was to show that the proposed preconditioning strategy leads to the number of GMRES iterations, needed for convergence, being independent of the problem size. In practice, the solution of (4.22) is only required as part of an inner iteration and so it is not generally necessary to solve it so exactly, i.e. it is only necessary to solve the linear system (4.22) to a sufficient precision in order to determine a good Newton update to achieve convergence of the non-linear system (4.21). Therefore, in the case of point contacts, the maximum dimension of Krylov-subspace is fixed to C (with no restarts) in order to approximately solve the linear system (4.22) at each Newton step. Our experience shows that the choice $C = 15$ works reliably for the EHL cases considered here, and prevents one from over solving the linearized system (4.22) at each of the Newton iterations. Again, as for the line contact problems, the initial guess used only consists of the Hertzian pressure profile for pressure and a small positive value for central offset film thickness (H_0). Finally, a Poisson ratio 0.3 has been used for the problems considered in this chapter whereas the Young's modulus for the equivalent non-dimensional elasticity problem is obtained according to equation (2.40). Moreover, the penalty parameter ξ is chosen to be $10^6 \times h_e^2$ as suggested in [51], where a different value of ξ is used for each element of characteristic length h_e. Moreover, as for the line contacts, the very small negative pressures in the cavitation regions are treated as zero while updating the density and viscosity.

Throughout this chapter meshes have been generated using the open source software NETGEN [95]. This allows meshes of a given local spacing to be defined, as well as permitting hierarchical mesh refinement to take place globally. Moreover, it allows one to perform mesh optimization over the meshes resulting from hierarchical mesh refinement. All timings reported here were computed using an AMD Opteron(tm) Processor 8384 @ 2.7GHz with a maximum of 128 GB RAM.

Table 6.1: Non-dimensional parameters for the contact between steel surfaces [104].

Parameters	Test Case 1	Test Case 2
Moes parameter, L	10	10
Moes parameter, M	20	200
Maximum Hertzian pressure, p_h	0.45GPa	0.97GPa
Viscosity index, α	$2.2 \times 10^{-8}\text{Pa}^{-1}$	$2.2 \times 10^{-8}\text{Pa}^{-1}$
Viscosity at ambient pressure, η_0	0.04 Pa s	0.04 Pa s
Total speed, u_s	$1.6\ \text{m}\,\text{s}^{-1}$	$1.6\ \text{m}\,\text{s}^{-1}$

6.3 Selection of Efficient Meshes

For the point contact problems considered here, the Reynolds equation needs to be solved on a 2D domain and the elasticity equation on a 3D domain. As previously mentioned, following Habchi [50], we let Ω represent a 3D domain $-30 \leq X \leq 30; -30 \leq Y \leq 30; -60 \leq Z \leq 0$. In [50] experiments were undertaken on different dimensions leading to the said dimension being adopted in their work. In this section, we first discuss the accuracy of the elastic deformation solution by defining suitable meshes and give a comparison of these against solutions obtained by using very fine meshes. These tests have been carried out first using a Hertzian pressure profile and then assuming a typical EHL pressure profile, though the results presented here are only for the EHL pressure profile. The test cases considered in this work are described in the Table 6.1. These test cases are taken from [104] in order to establish a comparison of the current model with a finite difference based model, so as to validate the calculated point contact results.

The term Ω_f is the part of the boundary of the 3D domain which corresponds to the contact region and is chosen to be of dimension $-4.5 \leq X \leq 1.5; -3 \leq Y \leq 3$. Note that three resolution cases, named Resolution 1, 2 and 3, are used in this work, corresponding to regular grids of 64×64, 128×128 and 256×256 points in Ω_f respectively.

It should be noted that the fine mesh cases used for comparison of solutions lead to very large computational problems and it has only been practical to solve these large problems with the development of proposed efficient preconditioned iterative solver.

6.3.1 Accuracy of the Elastic Deformation

The pressure generated inside the lubricant film is high enough in the contact region to lead to a significant elastic deformation while, on the other hand, the pressure outside the contact region is relatively low. Moreover, the precision of the computed elastic deformation is most important in and around the contact region, since this is where it has most effect on the Reynolds equation. Hence, we propose that a fine 3D mesh is required in the contact region up to a certain depth, but not in the whole elasticity domain. Neverthe-

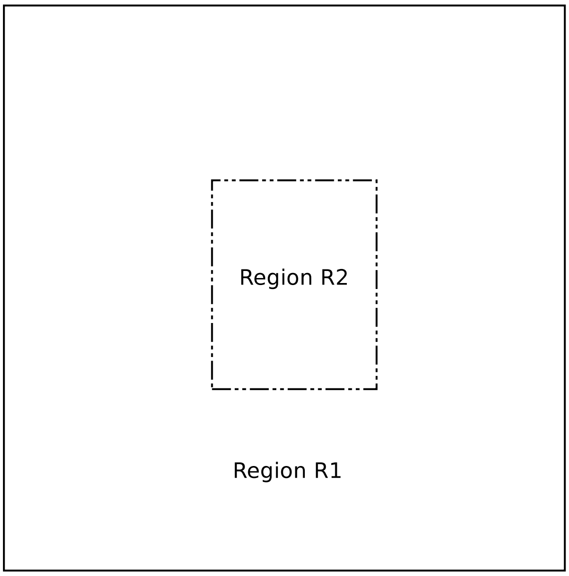


Figure 6.1: A view of the top of the 3D domain

less, a suitable mesh size needs to be defined outside the contact region in order to get a sufficiently accurate solution throughout the whole domain.

Here, we present selected meshes which give a sufficiently accurate elastic deformation solution at as low a cost as possible. In order to show the effectiveness of these meshes, the elastic deformation solution is compared against the solution obtained on very fine meshes. In the case of fine meshes, we split the 3D domain into two regions. The first region is of dimension $-4.5 \leq X \leq 4.5; -3 \leq Y \leq 3; -3 \leq Z \leq 0$ and the second region is the remainder of the domain. A mesh size corresponding to different resolution cases is adopted everywhere in the first region and a mesh size of 0.5 is used in the remaining region. A view (XY plane, where $Z = 0$) of the top of the domain is shown in Figure 6.1. For the sake of simplicity, the notations $R1$ and $R2$ are used to represent these two different regions in the domain, $R1$ representing the whole domain, and $R2$ representing the central region of dimension $-4.5 \leq X \leq 4.5; -3 \leq Y \leq 3; -3 \leq Z \leq 0$. It should be noted that the length of $R2$ is chosen $-4.5 \leq X \leq 4.5$ instead of $-4.5 \leq X \leq 1.5$ (see definition of Ω_f above) in order to get a sufficiently accurate elastic deformation solution. We carried out different experiments to ensure that any increase in the depth of $R2$ and any further decrease in the mesh size adopted in the remaining region does not lead to any significant improvement in the accuracy of the elastic deformation solution (see Appendix A for a more complete report of these computational experiments). Note that this two-region strategy for defining fine meshes for the three resolution cases considered leads to 3505403, 4950405 and 15802299 nodes in the elasticity mesh respectively.

Table 6.2: A set of mesh sizes defining the selected mesh for each resolution case.

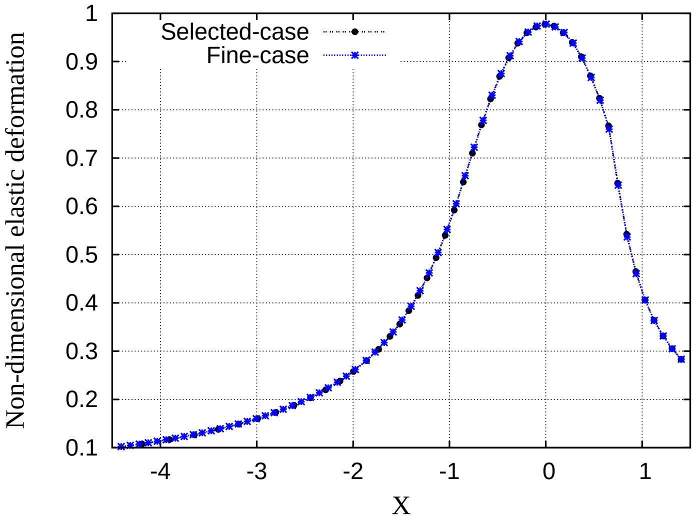
Selected Case	Mesh size 1	Mesh size 2	Mesh size 3	Mesh size 4
Resolution-1	0.09375	1.0	4.0	4.0
Resolution-2	0.04688	1.0	2.0	2.0
Resolution-3	0.02344	0.7	1.0	2.0

Table 6.3: Root mean square error for the displacement in all three selected resolution cases along with their fine cases: Test Case 1.

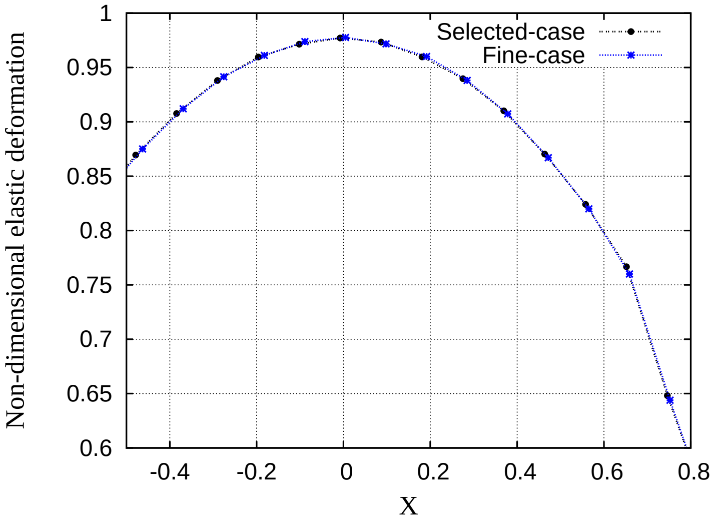
Case	Fine cases		Selected cases		Total reduction
	nodes	RMSE	nodes	RMSE	in the problem size (%)
Resolution-1	3505403	1.64×10^{-3}	97687	2.43×10^{-3}	97.2%
Resolution-2	4950405	6.25×10^{-4}	221260	9.85×10^{-4}	95.5%
Resolution-3	15802299	$-$	705860	4.80×10^{-4}	95.5%

Having established meshes to act as a benchmark for each of the three resolutions to be considered, we now seek to define meshes with fewer degrees of freedom that are able to maintain the required resolution in Ω_f and yield results of similar accuracy. To do this we split the 3D domain into four regions. The first two regions are hemispherical regions, centred on the origin, with radii ‘1.5’ and ‘15’ respectively. The third region is a cube of dimensions $-20 \leq X \leq 20; -20 \leq Y \leq 20; -40 \leq Z \leq 0$ while the fourth region is the remainder of the domain. The choice of the first region is based on the fact that this includes the region where the pressure values are most significant (especially in the inlet region). The choices of regions 2 and 3 are based upon a large number of different computational experiments that are reported in Appendix A. A selected resolution (Mesh size 1) is specified for all points inside the region 1 and a suitable mesh size (Mesh size 2) is defined for points on the curved boundary of region 2. For all other interior points of region 2, linear interpolation is used to define a local mesh size. It should be noted that in the remaining regions of the domain (outside of region 2), different large mesh sizes (Mesh size 3 in region 3 and Mesh size 4 in region 4) are used depending upon the resolution used in the most central region. The mesh sizes 1 to 4 (for the four regions respectively) for each resolution case are given in the Table 6.2, defining the proposed meshes. Note that the purpose of defining region 3 (particularly for resolution-3) is to control the huge increase in the number of nodes in the mesh without significantly affecting the overall accuracy of solution.

The root mean square error (RMSE) for the displacement in all three selected resolution cases, along with their fine (benchmark) cases, are calculated with respect to the fine case of resolution 3 and are given in the Table 6.3. These figures are obtained by comparing the computed solution on the given mesh against the best available solution, on the finest available mesh. One can see that the selected cases lead to a small fraction of the



(a)



(b) Zoom-in

Figure 6.2: Elastic deformation solution along the central line in Ω_f (where $Y = 0$): Fine and Selected case of resolution-1: Test Case 1.

nodes, less than 5%, compared to their corresponding fine cases (with the same resolution in the contact region) without significantly affecting the accuracy of the solution. In particular, for Resolution 1 and Resolution 2 the errors in the selected cases are of the same order of magnitude (less than a factor of 2 difference) as those based on the benchmark meshes of the same resolution. Moreover, a view of the elastic deformation solution using the fine and selected cases at resolution-1 along the central line in Ω_f (where $Y = 0$) is given in Figure 6.2, which also shows that both results are extremely close to each other.

Table 6.4: A comparison of point contact results over fine and selected meshes: Test Case 1.

This model (fine-meshes)				This model (selected-meshes)			
n_p	Total dof	H_c	H_m	n_p	Total dof	H_c	H_m
4450	10486550	0.42306	0.28921	1690	294056	0.42242	0.28603
17732	14834838	0.42999	0.29123	4854	666160	0.42842	0.28996
67350	47440138	0.43129	0.29202	16776	2131882	0.43059	0.29167

Table 6.5: A comparison of point contact results over fine and selected meshes: Test Case 2.

This model (fine-meshes)				This model (selected-meshes)			
n_p	Total dof	H_c	H_m	n_p	Total dof	H_c	H_m
4450	10486550	0.070251	0.028999	1690	294056	0.070361	0.027603
17732	14834838	0.078987	0.036286	4854	666160	0.078910	0.035759
67350	47440138	0.081202	0.038426	16776	2131882	0.081243	0.037901

6.3.2 Point Contact Results

In this subsection, we discuss the accuracy of the full EHL solution computed over the selected meshes which were determined, as described in the previous subsection, based upon numerical experiments to ensure a sufficiently accurate elastic deformation solution with a significantly reduced computational cost. For this purpose, a comparison of newly computed results is made with those obtained over the benchmark meshes (very fine) used in the previous subsection. Note that the latter meshes have already been used in Chapter 4 to validate the accuracy of point contact results against the previously published results using the integral approach in a finite difference based model [104].

Table 6.4 gives a comparison of these results for the Test Case 1, where n_p denotes the number of pressure unknowns. One can see that the accuracy of the solution is not significantly affected (relative to the overall discretization error) while keeping in mind the total decrease in the size of problem. For example in case 3 of Table 6.4, a reduction of 45M degrees of freedom (dof) leads to a very small drop in the accuracy of the solution. The total relative difference between the two solutions is 0.16% and 0.12% for the central film thickness (H_c) and the minimum film thickness (H_m), respectively. In particular, in the selected cases, the same order of magnitude in error is observed as with the fine cases.

A similar trend in the solutions is observed for Test Case 2 which is a comparatively more heavily loaded case. Table 6.5 provides a comparison of fine case results with those computed over the selected meshes. Again it can be seen that the accuracy of the solution is not significantly affected for this loaded case (Test Case 2) while computing the solution over the selected meshes. The total relative difference between the two solutions is 0.05% and 1.4% for the central film thickness and the minimum film thickness, respectively.

This experiment provides further evidence that the mesh selection approach proposed in this work may be applied across a range of EHL cases.

6.4 Performance of Solvers

In this section the performance of preconditioned iterative solvers are compared to the sparse direct solver. To solve the linearized system (4.22) at each Newton iteration, again different linear solvers are used. For the sparse direct results UMFPACK [28] is used, while in the iterative solvers the right-preconditioned GMRES method is used. In the iterative solvers, again preconditioning of the elasticity block (J_{22} in (4.23)) is done using algebraic and geometric multigrid. Contrary to the line contact case, the Reynolds block is not tridiagonal, therefore we have used a sparse direct solver (UMFPACK) for this block of the preconditioner. The choice of the sparse direct solver is inspired from the fact that it performs very well for the 2D EHL problem both in terms of memory usage and computational time.

6.4.1 Algebraic Multigrid Solver

Before discussing geometric multigrid preconditioning we first give a comparison between the performance of the sparse direct and the preconditioned iterative solvers, with AMG preconditioning of the elasticity block. These results are obtained using the moderately loaded Test Case 1 (see Table 6.1) and are given in Table 6.6 in terms of total iterations, total time, time per nonlinear iteration and memory usage. It can be seen that the AMG solver is performing very well compared to the sparse direct solver, both in terms of memory usage and time. Furthermore, both memory usage and computational time is growing almost linearly with the increase in the problem size. For this 3D case the sparse direct solver soon reaches its limitations in terms of both the memory and the CPU time required (due, we believe, to the much greater fill-in that occurs for the 3D problem relative to the 2D line contact case). Indeed, we have insufficient memory to compute the finest case. The preconditioned iterative approach is far superior.

Table 6.6: Comparison of sparse direct and iterative (AMG) results, Test Case 1

n_p	Total dof	Sparse Direct Solver				Prec. Iterative Solver (AMG)			
		ni	t (s)	t/ni	mem.(Gb)	ni-li	t (s)	t/ni	mem.(Gb)
1690	294056	9	20685	2298.33	21	9-135	729	81.00	0.7
4854	666160	7	68567	9795.29	56	7-105	1242	177.43	1.5
16776	2131882	–	–	–	–	7-105	4184	597.71	4.9

Table 6.7: Comparison of preconditioned iterative point-contact results (with AMG and GMG) for cases Test Case 1 & Test Case 2.

n_p	Total dof	Prec. Iterative Solver (AMG)				Prec. Iterative Solver (GMG)				
		ni-li	t (s)	t/ni	mem.(Gb)	ni-li	t (s)	t/ni	mem.(Gb)	levels
Test 1										
4675	529214	7-105	1013	144.71	1.2	7-105	814	116.29	0.8	3
18837	4152538	7-105	9046	1292.29	9.9	7-105	6894	984.86	6.0	4
Test 2										
4675	529214	12-180	1524	127.00	1.2	12-179	1213	101.08	0.8	3
18837	4152538	11-165	11930	1084.55	9.9	12-180	10098	841.50	6.0	4

6.4.2 Geometric Multigrid Solver

As for the line contact problems considered in the previous chapter a further time and memory usage reduction is possible if AMG preconditioning of the elasticity block is replaced with GMG preconditioning on a suitable mesh. Such a mesh sequence is obtained through uniform refinements of the coarsest mesh. The performance of GMG preconditioning depends upon the coarsest grid used and how efficiently the coarsest grid problem is solved. In this work UMFPACK is used to solve the coarsest grid problem. In order to obtain the efficient coarsest grid solution, the linearity of the elasticity block (J_{22} in (4.23)) is taken into account. In other words, LU factorization of the coarsest grid matrix is done only once and these factors are stored in order to avoid the repeated factorization and to use them consistently in the later computations. It should be noted that linear tetrahedral elements have been used in the meshing of the elasticity domain. Starting with a suitable coarse grid with a total of 3042 mesh points, each tetrahedral element is divided into 8 child tetrahedra (e.g. see [84, 96]) at each refinement level, growing the number of elements eight times at each refinement level. This is the main drawback of GMG preconditioning: that we have to accept this restriction over the meshes. Moreover, the division of each tetrahedron is done in two phases, the first leading to the removal of the four corners, leaving an octahedron behind. This octahedron is further divided to produce four new tetrahedra. There are three choices of diagonals which can be used to divide this octahedron. In this work, we have used the largest diagonal [69, 96], however other choices [84] are possible. It should be noted that accuracy of the solution depends upon the quality of the meshes generated at each level (which is discussed in detail later in this chapter). Our main concern in this section is to discuss the performance of AMG and GMG preconditioning however. Table 6.7 gives a comparison of the AMG and GMG preconditioning strategies for Test Case 1 and Test Case 2. The efficiency and optimality of both solvers is apparent as the time growth and the memory usage is almost linear. Again, a further reduction in time and memory usage is achieved by using GMG

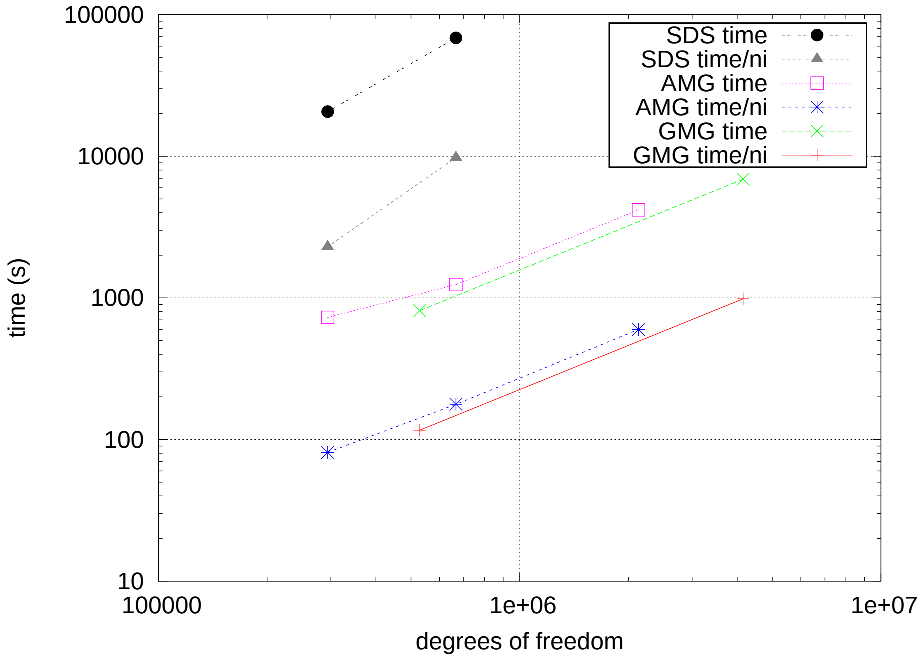


Figure 6.3: Performances of different point contact solvers. Test Case 1

preconditioning of the elasticity block.

The overall performance showing total time and time per nonlinear iteration for the different solvers for Test Case 1 can be seen in Figure 6.3. One can see that the preconditioned iterative solvers substantially outperform the application of a sparse direct solver. It is also clear that the AMG performance is very close to GMG and that, as for the line contact case, it is not likely that the benefits of the GMG approach are sufficient to overcome the mesh restrictions compared to AMG.

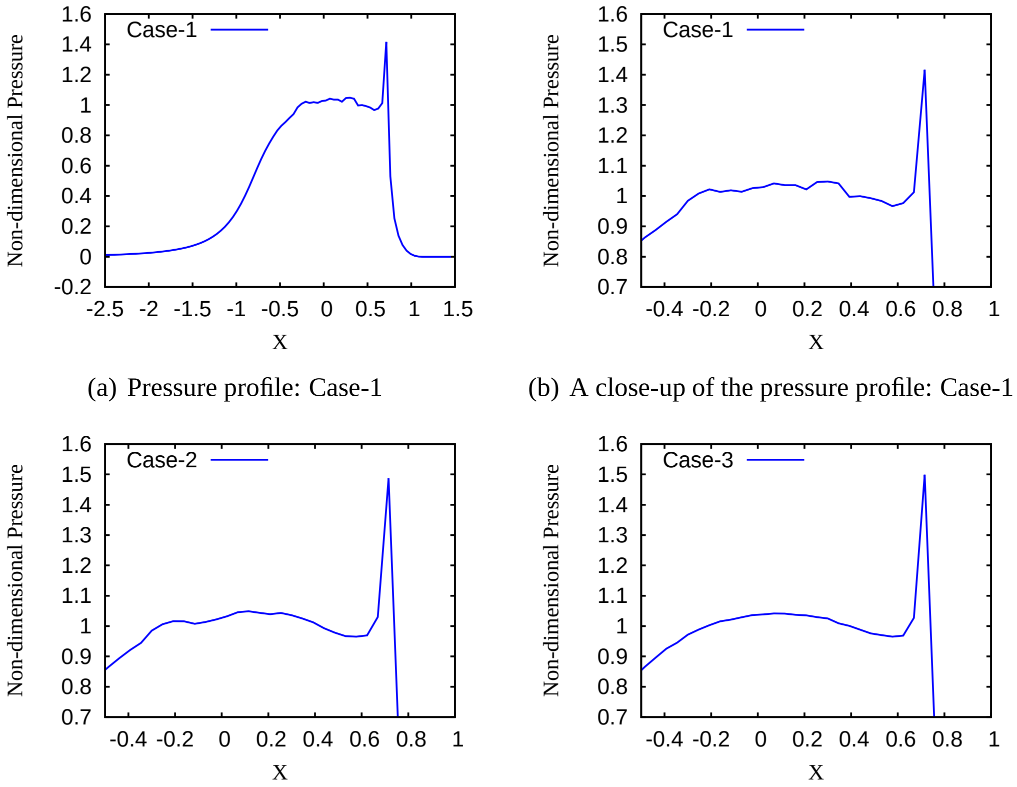
Consider, for example, the finest case in Table 6.6, the unstructured mesh has $n_p = 16776$ with 2131882 degrees of freedom in total. For an equivalent resolution in Ω_f ($n_p = 18837$ in Table 6.7) the GMG solver leads to comaparatively twice the problem size (i.e. 4152538 degrees of freedom in total). In this specific example, the AMG solver (with least problem size) leads to a saving of about 40% in the CPU and 20% in the memory usage compared to the GMG solver (with a large problem size). In addition to the mesh restriction, the quality of hierarchical meshes used in the GMG solver is generally not good enough compared to the unstructured meshes used in the AMG solver. This issue is discussed in detail in the following section.

6.5 Hierarchical Meshes

In addition to the local mesh resolution, another important factor which affects the accuracy of a solution of a problem is the quality of mesh used [14, 15]. In other words, optimizing the quality of a mesh prior to computing the solution of a problem can improve the accuracy of the solution. In this subsection, this aspect of the solution process is considered by computing the EHL solution over different meshes of resolution 2. In mesh optimization, various heuristic measures of the quality of the mesh are improved by modifying the mesh in different ways. The mesh optimization in NETGEN [95] includes local reconnection via edge and face swaps, local node movement, and some collapsing of elements. The metric used is based upon minimizing an error functional which quantifies the quality of the mesh. The Netgen [95] optimization process distinguish between the metric optimization and the topological optimization. In the former case, mesh quality is increased with points movement. Once the quality does not improve any further with point movement then topological changes are made in the mesh where some elements are removed and the points are connected in a new manner. Moreover, edge and face swaps are also performed to improve the mesh quality. For more details about the optimization process in NETGEN, see [95].

For this work an experiment is carried out for the Test Case 1 in which an optimized coarse mesh (generated by NETGEN) is chosen as an initial mesh. The choice of this initial mesh is made such that the two levels of uniform refinement gives an equivalent mesh to that with selected mesh case of resolution 2 (see Section 6.3). Three cases are then considered. In the first case (Case 1), two levels of uniform refinement are applied to the initial mesh. Each refinement involves the division of each tetrahedral element into 8 child tetrahedra, growing the number of elements eight times at each refinement level. The division of each tetrahedron is done in two phases, the first leading to the removal of four corners, leaving an octahedron behind. This octahedron is further divided to produce four new tetrahedra. There are three choices of diagonals which can be used to divide this octahedron. In Case 1, we have used the longest diagonal [69, 96], however other choices [84] are possible. In the second case (Case 2), the two levels of refinement are carried out using NETGEN [95]. Among the three choices of diagonal to divide an octahedron the NETGEN chooses the best one. Finally, an optimization is performed (within NETGEN [95]) over the mesh obtained in Case 2 leading to Case 3.

It may be observed that Case 1 leads to a poor quality solution since unphysical oscillations are clearly visible in the pressure solution, see Figure 6.4. On the other hand, Case 2 yields a slightly better solution (with fewer oscillations) than that of Case 1, while



(a) Pressure profile: Case-1

(b) A close-up of the pressure profile: Case-1

(c) A close-up of the pressure profile: Case-2

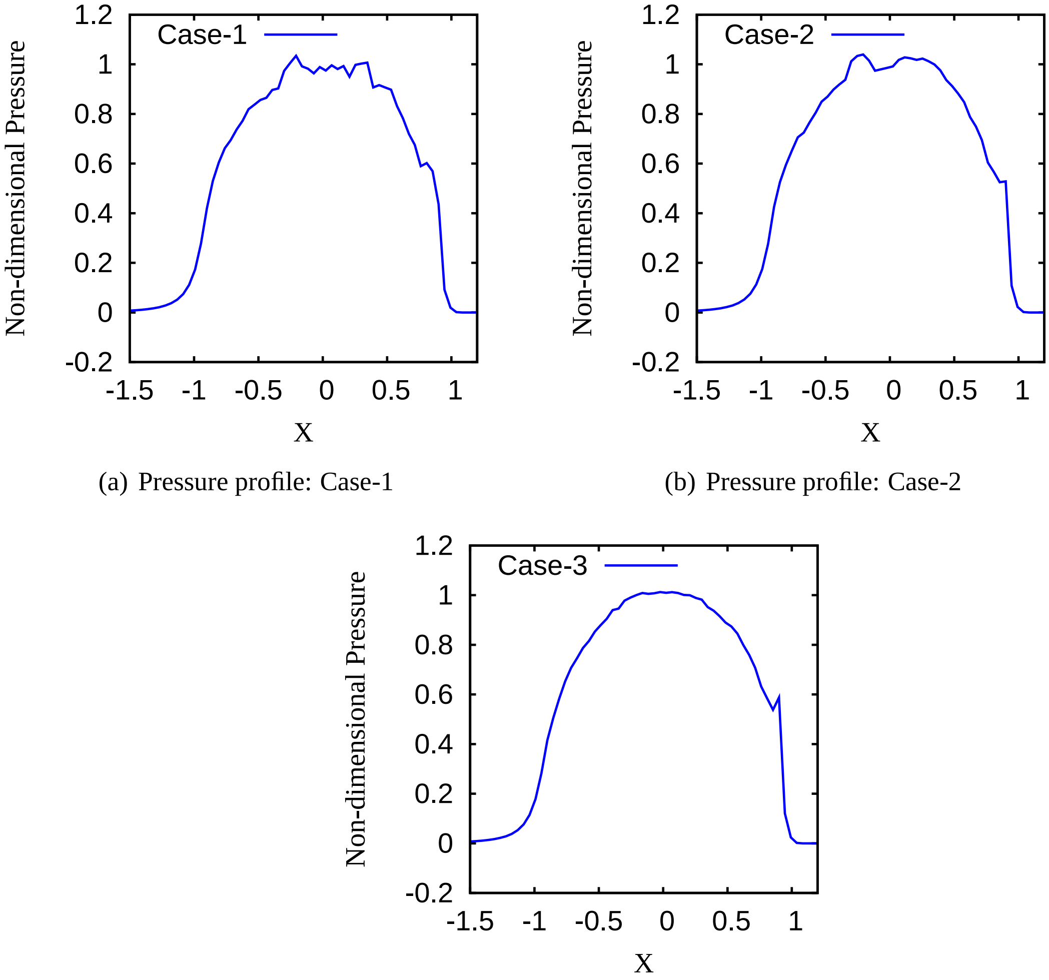
(d) A close-up of the pressure profile: Case-3

Figure 6.4: The shapes of pressure profile along the central line (where $Y = 0$) computed over optimized and non-optimized meshes: Test Case 1

Case 3 leads to a smooth pressure profile (as smooth as could be achieved at this spatial resolution). It should be noted that the number of degrees of freedom for the first two cases are the same (710906), as a result of the uniform refinement procedures, while in Case 3 this number is 670577. This slight reduction is due to the collapsing of nodes, edges, faces and elements that takes place during the mesh optimization process [95]. The key observation however is that the smoother, optimized, mesh can yield better qualitative accuracy in the solution than that of a non-optimized mesh with more nodes. Recall that, in this example we have selected the initial mesh such that two levels of refinement produce an equivalent mesh to the selected case of resolution 2 (see Section 6.3). In order to assess the accuracy in a more quantitative manner the RMSE of the pressure and film thickness solutions for these different resolution 2 cases are calculated with respect to the fine case of resolution 3, and are given in Table 6.8. It can be seen that the RMSE of both the film thickness solutions and the pressure solutions are very similar to those for the selected mesh case. The accuracy of the pressure profile increases while switching

Table 6.8: Root mean square error of pressure and film thicknesses solutions of various resolution 2 cases with respect to the fine case of resolution 3: Test Case 1.

Case	dof	Pressure RMSE	Film thickness RMSE	t(s)	t/iter	mem.(≈Gb)
Case 1	710906	3.27×10^{-2}	3.10×10^{-3}	1421	203.0	1.6
Case 2	710906	2.71×10^{-2}	3.31×10^{-3}	1397	199.57	1.6
Case 3	670577	2.42×10^{-2}	3.48×10^{-3}	1343	191.86	1.5
Selected mesh (Res. 2)	666160	2.56×10^{-2}	3.12×10^{-3}	1242	177.43	1.5
Fine mesh (Res. 2)	14834838	1.39×10^{-2}	1.68×10^{-3}	31681	4525.86	34



(a) Pressure profile: Case-1

(b) Pressure profile: Case-2

(c) Pressure profile: Case-3

Figure 6.5: The shapes of pressure profile along the central line (where $Y = 0$) computed over optimized and non-optimized meshes: Test Case 2

from Case 1 to Case 3 and in Case 3 it is even slightly improved over that of the selected case. Moreover, it is apparent that the accuracy of the pressure and film thicknesses solutions is not substantially affected as compared to the solutions computed on the very fine reference mesh with a huge number of degrees of freedom (14834838).

A similar behaviour in the accuracy of the solution is observed for heavily loaded Test Case 2. Figure 6.5 shows the pressure profiles (along the central line where $Y = 0$) for each of three mesh cases. Again Case 1 leads to a poor quality solution while Case 2 yields a slightly better solution. As for the Test Case 1, Case 3 (optimized mesh) leads to a comparatively more smooth pressure profile than Case 1 and Case 2. However, the pressure profile is not completely smooth for this spatial resolution but it can be further smoothed with an increase in the spatial resolution. Nevertheless, the results shown provide a further evidence that the accuracy of an EHL solution can be significantly improved using a post processed optimized mesh.

6.6 Conclusion

In this chapter we investigate the accuracy of the elastic deformation solution, as well as the complete point contact EHL results, over a number of different finite element meshes for the linear elasticity problem. It has been possible to show that a judicious choice of these finite element meshes can allow a substantial reduction in the total number of degrees of freedom without reducing the overall accuracy too significantly relative to the discretization error in the Reynolds equation solution. In the specific example included here this results in a reduction in CPU of about 96% (from $31681s$ to $1242s$) and a reduction in memory of about 96% (from 34Gb to 1.5Gb) for the AMG solver. The selected elasticity meshes presented in this chapter have been found to be accurate over a relatively small range of EHL problems, however, it requires a systematic study over a wider range of cases to demonstrate, its applicability in full. Moreover, for harder materials, a comparatively smaller mesh size (away from the contact region) may be needed to obtain appropriate meshes using the same methodology so as to achieve a desired accuracy in a solution. Note that if the surface roughness is taken into account then a finer mesh in the contact region (high pressure region) may be required and this may therefore affect the mesh sizes close to the central contact region accordingly. Nevertheless the mesh approach used here would still be applicable.

Furthermore, we discuss the performance of different fully coupled EHL point contact solvers by giving a detailed comparison of their computational times and memory costs. For the point contact problems, presented in this chapter, the application of our preconditioning strategy outperforms the sparse direct solver very significantly, with huge savings in memory and time being achieved. Perhaps most importantly the growth in both time and memory for the preconditioned iterative approach appears to be linear.

Finally, it has been shown that the performance of the AMG approach is very close to GMG, and that the benefits of the GMG approach are not enough to overcome the mesh restrictions compared to AMG. A further observation provided in this work is that the use of unstructured hierarchical meshes without appropriate mesh optimization can lead to poor quality EHL results. In other words, optimizing a mesh prior to computing an EHL solution significantly improves its accuracy. As the surface mesh remains unchanged during the optimization process in NETGEN [95], the only reason for the improvement in the accuracy of an EHL solution is the improved accuracy in corresponding elastic deformation solution. In short, an AMG solver with least problem size yields a better accuracy in an EHL solution more economically than a GMG approach.

The effect of mesh quality over the accuracy of an EHL solution is also likely to be an important observation for future work where we seek to make use of local error estimation to control the mesh refinement locally in order to automate the generation of the linear elasticity finite element meshes (see the following chapter for an initial investigation of this approach).

Chapter 7

An Adaptive Method for EHL Problems

7.1 Overview

Recall from previous chapters that a fully-coupled EHL point contact problem involves solving a linear elasticity equation on a 3D domain, and the Reynolds equation on a small part of its surface boundary (the so-called the fluid domain). A large number of experiments were carried out to define appropriate 3D meshes (corresponding to different mesh resolutions in the contact region) with a view to obtaining a satisfactory EHL solution at the lowest cost as possible. Note that, in the selection of those meshes, smaller mesh sizes were used only in the contact region where the pressure solution exhibits the large variation [45, 70].

In this chapter, the development of a locally adaptive finite element solution scheme for fully-coupled EHL point contact problems is discussed, which automatically refines the mesh in the fluid as well as the elasticity domain (where the fully-coupled solution exhibits large variations). In Section 7.2, a description of the proposed adaptive algorithm is provided. Section 7.3 discusses in detail an 'a posteriori' error assessment which is used to find the local, as well as global, error estimates. In Section 7.4, different refinement criteria are described to target a list of elements for refinement. Section 7.5 provides details about the refinement algorithm which is used to carry out the mesh refinement process. Section 7.6 provides a description of a procedure for post-optimization of the meshes generated. The layout of different variants of the adaptive solver developed in this work is presented in Section 7.7. Finally, in Section 7.8, numerical results are presented to show the effectiveness and the performance of the proposed procedure for solving EHL problems.

7.2 Mesh Adaptivity

The accuracy of a finite element approximation depends, amongst other things, upon the computational mesh being considered. The use of finer meshes (i.e. with smaller elements) is one way to improve the accuracy of a finite element approximation. In other words the overall difference (error) between a computed and an exact solution can be reduced with the use of finer meshes. The case of uniformly refined meshes often leads to an excessive increase in the size of the discrete problem, and it becomes very expensive, from a computational point of view, to achieve a desired accuracy. In particular the use of very fine elements *everywhere* in the computational domains is not often required (see Chapter 6). Specifically it may only be required to use the finest elements in those regions where the local error is greatest, such as regions where the solution has sharp features, e.g. steep gradients, singularities or discontinuities, etc. The errors arising from such regions make a major contribution to the overall global error, and the accuracy of a solution can be improved by paying particular attention to such regions, based on an assessment of the local error (through an error estimate or indicator).

This goal can be achieved using an adaptive procedure [69, 74, 96, 109, 110] which seeks to automatically optimize the computational process so as to obtain the desired performance results (i.e. global accuracy) at a minimal computational cost. Different adaptive procedures are possible but they typically involve applying local refinement/derefinement and/or adjusting the order of approximation of the method in the regions where the large errors come from. The various strategies for controlling the finite element adaptivity generally fall into three categories:

- h-adaptivity [67,69,87,92,96] consists of using the same degree of elements throughout, but the sizes of elements are changed locally to improve the accuracy or efficiency. In such a case the elements which show large errors in their solution are divided into smaller elements.

- p-adaptivity consists of using the same number of elements but the order of approximating polynomials are increased to achieve a desired accuracy in a computed solution. This is typically used in conjunction with h-adaptivity [12, 31, 65].

- r-refinement keeps the number of nodes and the order of the elements fixed but adjusts their position to improve the accuracy of the solution. Examples include [9, 74, 75].

As noted above, these strategies may also be applied in combination, for example hp-refinement combines both the h-refinement and the p-refinement (i.e both the local mesh sizes and order of polynomial are altered) in an efficient manner to obtain best results. Moreover, h-refinement can be applied in different ways:

- local refinement: elements showing large errors are divided into smaller elements [67, 69, 96].
- mesh coarsening: this involves coarsening the mesh in regions where the solution on the elements exhibits small errors. This is done in order to reduce the computational cost [67, 69, 92, 96, 109]. However the efficiency of the algorithm may be affected due to complex data management.
- re-meshing: in this case new local mesh sizes are determined on the basis of a computed solution, and a totally new mesh is generated again [58, 86, 109]. The advantage is that both the refinement and derefinement can easily be implemented. However the drawback is the difficulty of transferring the solution between different grids, and it may also be expensive to generate a totally new mesh especially in 3D cases.

An adaptive algorithm generally involves the followings steps [74]:

1. An input initial mesh.
2. Solve the corresponding system of discrete equations.
3. Compute the local error estimates or indicators for each element in the mesh.
4. If the errors are in the prescribed limit then the process is complete, otherwise identify a list of elements exhibiting large or small errors.
5. Perform the refinement process, and goto step 2.

In this work, only the use of h-adaptivity is considered with first order finite elements. The reason for this is so as to make use of the preconditioned iterative solver (developed and used in the previous chapters) at each adaptive level in order to obtain an efficient solution at step 2 of the above algorithm. Note that derefinement has not been used, and a similar effect to that with remeshing will be discussed in Section 7.8. In the following section, a detailed discussion is provided about error estimates, refinement criteria and the refinement process.

7.3 A Posteriori Error Assessment

The error assessments generally fall into two categories: ‘a priori’ error assessment and ‘a posteriori’ error assessment. An ‘a priori’ error assessment takes into account advance knowledge of the behaviour of the exact or the numerical solution or of input data, and is used where the knowledge of the specific numerical solution is not required. For example, where the solution is known to possess a singularity an optimal refinement strategy may be determined in advance, e.g. [7]. On the other hand ‘a posteriori’ error assessment is generally based on the computed numerical solution and is therefore an important ingredient for an adaptive finite element procedure. There are many ‘a posteriori’ error estimators developed, e.g. [3, 54, 110], which generally fall into two categories: recovery based error estimators [110] and residual based error estimators [3]. The work presented in this chapter only takes into account error estimators of the first type. To explain such estimators, let us assume that $\mathbf{u}_h$ is a finite element approximation to an exact solution $\mathbf{u}$ of an elasticity equation then the error in the computed solution is the difference:

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_h,$$

and the error in their corresponding gradients or stresses, denoted by σ, is:

$$\mathbf{e}_\sigma = \sigma - \sigma_h.$$

For an elasticity problem, stresses are calculated from the finite element solution by:

$$\sigma_h = \mathbf{DSu}_h,$$

where the elasticity matrix $\mathbf{D}$ and the differential operator $\mathbf{S}$ are given by [4] (for the 3D problem):

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix}.$$

Instead of the point-wise definition of errors, an integral measure is often used to present errors throughout an element or the whole domain (Ω). In the energy norm, a common ‘a posteriori’ error estimate, based on the stresses of the solution of an elasticity problem, takes the following form [110]:

$$\|\mathbf{e}_\sigma\|^2 = \int_\Omega (\sigma - \sigma_h)^T \mathbf{D}^{-1} (\sigma - \sigma_h) \mathrm{d}\Omega.$$

Since neither the exact solution u nor σ are always in hand, a reasonable error estimator can be obtained if the true gradients σ are replaced with a suitable approximation σ^* (obtained by a suitable post-processing of the finite element approximation):

$$\|\mathbf{e}^*_\sigma\|^2 = \int_\Omega (\sigma^* - \sigma_h)^T \mathbf{D}^{-1} (\sigma^* - \sigma_h) \mathrm{d}\Omega \,. \tag{7.1}$$

Generally the gradients computed from the finite element approximation are discontinuous over the inter-element boundaries. An approximation is then made at each node by averaging the elemental contribution of such gradients in a patch of elements sharing that node. It is then possible to use the interpolating polynomials (the same as those used in the finite element approximation) to define a continuous, recovered, approximation on the whole domain. Such class of methods are often known as averaging method [3]. Various estimators can be distinguished based on the specific steps involved in the construction of the average or recovered gradients.

A well known post-processed, or recovery based, error estimator was proposed by Zienkiewicz et al. in late ‘80s [110] (known variously as Zienkiewicz-Zhu or ZZ or Z^2 error estimator). Later on, the authors presented an improved estimator based on superconvergent patch recovery [111, 112]. These estimators are based on the fact that the approximated solution is less accurate at the element’s nodes and boundaries. However, there are points within the elements where the gradients are more accurate and converge to exact values more quickly as the element size decreases. Specifically, such points often exhibit superconvergent behaviour in the solution and are therefore referred to as superconvergent points. Thus a more accurate estimate (σ^*) of the true gradient (σ) is recovered at a node by interpolating between the gradients at the superconvergent points in a patch of elements surrounding that node. The ZZ error estimator is economical and easy to implement. Furthermore, the ZZ error estimator has been shown to be effective compared to other residual based error estimators in different comparative studies, see for example [4, 6, 8].

Finally, the norm used above is defined over the whole domain Ω. In practice, the squared value of the norm can be obtained by summing up the individual element contributions, i.e.

$$\|\mathbf{e}_\sigma^*\|^2 = \sum_{i=1}^{N} \|\mathbf{e}_\sigma^*\|_i^2, \tag{7.2}$$

where i is the element number and N is the total number of elements in a current mesh.

7.4 Refinement Criterion

In the previous section, a recovery based error estimator is discussed to determine an approximation to the global, as well as local, error produced in a finite element approximation. If the global error is already within the prescribed bounds for a given mesh then the goal is already achieved. However, this is often not the case and refinement is typically necessary in all or some parts of the domain, which exhibit large or unacceptable errors. In practice, a tolerance (η_{tol}) is usually specified for the target relative error (η) in the final solution (or gradients), i.e.

$$\eta = \frac{\|\mathbf{e}_\sigma\|}{\|\sigma\|} \leq \eta_{tol} \ . \tag{7.3}$$

The refinement, solution and error estimation steps are repeated until this criterion is satisfied. If neither the true error nor the exact gradients are known then they are replaced with their best approximations available. Some times it is not possible to reach the prescribed limit (let say $\eta_{tol} = 5\%$ [109]) of permissible error (especially for the 3D problems) due to nature and availability of computer resources e.g. memory, speed etc. Therefore other alternatives such as maximum refinement levels, minimum element size, memory usage, etc. can be specified as stopping criteria. In this work, the maximum number of refinement levels are used as a stopping criterion for the adaptive procedure.

As stated earlier, refinement is necessary in the regions of largest error. In other words, one main requirement for an optimal mesh is that the error is equally distributed among all the elements in a mesh [78], i.e. the requirement:

$$\|\mathbf{e}_\sigma^*\|_k \leq \eta_{tol} \left(\frac{\|\sigma_h\|^2 + \|\mathbf{e}_\sigma^*\|^2}{N} \right)^{\frac{1}{2}} = e_{tol},$$

must hold for all the elements (numbered k) in a current mesh. In the above equation, N is the total number of elements and e_{tol} (average element error) represents the maximum permissible error for an element. In other words, the ratio:

$$\xi_k = \frac{\|\mathbf{e}_\sigma^*\|_k}{e_{tol}} > 1 \tag{7.4}$$

specifies the set of elements to be refined. Derefinement is also possible, to save computations, whenever $\xi_k < \xi_{deref} \ll 1$.

Another refinement criterion (used in this work) is based upon finding the maximum error in elements (e_{max}) and targeting elements for refinement according to the equation:

$$e_{tol} = ce_{max}, \tag{7.5}$$

where c is a constant (and different values of this constant are used in this work for demonstration purposes). Any decrease in this parameter may result in flagging quite a lot more elements for refinement and the required goal may not be achieved due to an excessive number of elements in the final mesh.

7.5 Tetrad

The mesh refinement algorithm TETRahedral ADaptivity (TETRAD) is a mesh adaption algorithm developed at University of Leeds by Speares et al. [96] in late '90s. A general description of this algorithm is provided in this section however, for more details one is referred to the original text [96]. The algorithm used in TETRAD is hierarchical in nature and is suitable for meshes consisting of tetrahedral elements. The mesh adaption algorithm is supported both by the mesh refinement and derefinement process. Assuming a good quality input mesh the refinement of the base level mesh takes place by addition of new nodes by edge, face and element subdivision and the changes in the mesh are kept track of via the construction of a data hierarchy. Contrary to refinement, the derefinement process is a process where nodes, edges, faces and elements are removed from the mesh to restore the original element(s) back. Note that the derefinement algorithm works only until the elements of the base level mesh are recovered back, and hence no further derefinement is possible.

The mesh refinement process is invoked automatically in a specific region of the computational domain depending upon where the corresponding solution has large estimated error and/or a larger spatial resolution is required. It is assumed that this information has been provided to TETRAD by a user subroutine in which the edges are either marked for refinement or derefinement (or no action). If an edge is marked for refinement then it leads to refinement of all elements sharing that edge. Similarly if an edge is marked for derefinement then all the elements which are sharing that edge are potentially derefined. The refinement process takes into account only two types of subdivisions. A regular subdivision in which each parent element is divided into eight child elements by introducing new nodes bisecting each edge. In the first instance this leads to removal of four corners leaving an octahedron behind. The division of this octahedron further results into four new child elements on the basis of dissection by the longest diagonal [69, 96]. The other kind of subdivision, the so-called green refinement, takes place where not all of the edges of an element are marked for refinement, and this avoids the possibility of introducing "hanging nodes" (nodes on edges or faces which are potentially not the vertices of all elements sharing those edges or faces) without introducing any additional edge refinement. Note that green refinement often leads to poor quality elements, and therefore a precaution is taken into account in the development of TETRAD that a green element may not be refined further. In such a case, the previous green refinement of the parent element is replaced with the regular refinement. Thus the green refinement always appears at the interface between lower and higher grid resolutions. As a consequence, the poor quality elements never appear in the region of interest provided appropriate flagging criteria have been used for adaption.

Finally, the developers showed that the scaling behaviour of the fundamental refinement process is close to optimal linear behaviour [96] and is not significantly affected by the mesh depth.

7.6 Optimization of Meshes

In the previous chapter, it was observed that the unstructured meshes resulting from hierarchical mesh refinement often lead to poor quality EHL results without appropriate mesh optimization. In other words, the accuracy of the EHL solution can be improved by optimizing the quality of mesh prior to any computation. In this work, this fact is also taken into account for the meshes resulting from the local refinement process.

In order to combine optimization with local mesh refinement, the meshes obtained once the refinement is performed are passed to NETGEN [95], where a smoothing process is performed via edge and face swaps, local node movement, and some collapsing of elements. Note that, unlike [74], the optimization does not seek to reduce the error further, rather it is undertaken to ensure minimization of a quality functional which quantifies the quality of the mesh. An advantageous side-effect of the optimization is that the collapsing of elements in the optimization process also leads to a reduction in the size of problem compared to the original mesh. In other words, this method has a resemblance with r-refinement cases, along with a possible reduction in the number of mesh points. A difficulty encountered with this type of approach is to handle transfer of the solution data between the grids. Furthermore, the optimization processes destroys the mesh hierarchy, so that neither de-refinement nor the use of GMG preconditioning is possible.

Smoothing via NETGEN [95] also has the drawback that the mesh optimization only takes place in the interior of the domain, i.e. the surface mesh remains unchanged. The advantage of this is that the pressure solution can be transferred to the new optimized mesh without any difficulty. However, to produce an initial guess for the elasticity solution on this changed mesh, one needs to solve the elasticity equation corresponding to the surface pressure. Hence, at the cost of a solution of the elasticity equation (equivalent to less than one fully-coupled iteration) one yields an initial guess for which the fully-coupled iteration converges very quickly. Note however that the next refinement of green 2D elements on the fixed surface mesh will lead to even more poor quality surface mesh elements, regardless of an optimized 3D mesh. The poor quality surface mesh in the fluid region may affect the accuracy of the pressure solution. One possibility to avoid the low quality surface mesh is to perform the mesh optimization only at the final level to improve the accuracy of the final solution. This is therefore considered as one of the possible strategies in the following section.

7.7 Solver Layout

In this section we discuss the overall layout of the adaptive algorithm used in this work. A suitable initial mesh is first generated using NETGEN [95], where a fine mesh is used in the contact region compared to the other parts of the domain. The choice of an initial mesh is made such that a reasonably good starting solution could be obtained. The main algorithm used in this work can then be split into the following steps.

1. Pass an initial mesh to TETRAD [96] to read-in the mesh and build all the data and their structures.

2. Construct lists of leaf elements and edges, and assign boundary conditions.

3. Set up and solve the fully-coupled EHL problem using the solver based on AMG preconditioning of the elasticity block (see Chapter 6).

4. Estimate the error within each element. If the maximum level is reached then output is produced and code exits, otherwise a list of elements is created for adaption (h-refinement).

5. Perform h-refinement within the TETRAD. For mesh optimization, goto step 6, otherwise goto step 2.

6. Optimize the locally refined mesh using NETGEN [95]. Free up all the previous data and structures except the new mesh and the solution data, and goto step 1.

Having defined the basic algorithm, a description of different variants of this adaptive algorithm is now provided. Recall from the previous section that the post processing (smoothing) of the adapted mesh may lead to even more accurate results. Note however that if the optimization in step 6 is performed then it destroys the mesh hierarchy. Moreover, calling the step 6 at each refinement level may result in a bad quality surface mesh after a number of levels, which may affect the accuracy of the solution of the Reynolds equation. To assess the accuracy of the solution, three possibilities are considered, which ultimately lead to three variants of the main algorithm.

- The first variant of solver skips the step 6 and repeats from step 2 until the maximum level criterion is reached. In this case TETRAD keeps a record of all of the refinement history and therefore green elements are prevented from further refinement (and the use of the GMG preconditioner is possible in theory too, though not implemented here) and the initial guess at each stage is a simple interpolant from the previous solution.

- The second variant of the main solver utilizes step 6 at each refinement level and therefore repeats the process from step 1 with new mesh. Since the surface mesh does not change, and hence the 2D fluid mesh, so the solution of the Reynolds equation is transferred to this new mesh without any difficulty, and solving the elasticity equation yields an initial guess for the displacement for this new mesh. Hence, an overall improved initial guess leads to fewer Newton iterations to achieve the convergence of the fully-coupled system. However, the quality of the surface mesh deteriorates with each additional local refinement.

Table 7.1: Non-dimensional parameters for the contact between steel surfaces [104].

Parameters	Values
Moes parameter, L	10
Moes parameter, M	20
Maximum Hertzian pressure, p_h	0.45GPa
Viscosity index, α	$2.2 \times 10^{-8}\text{Pa}^{-1}$
Viscosity at ambient pressure, η_0	0.04 Pa s
Total speed, u_s	$1.6\ \text{m}\,\text{s}^{-1}$

- To avoid the risk of successive green refinement at the surface mesh, the third variant only utilizes the step 6 at the final level of refinement, and hence a surface mesh is obtained with a relatively good quality.

In the following section, a comparison of accuracy and the performance of these variants is provided to assess the different strategies used in the above variants of the solver.

7.8 Numerical Results

In this section, a comparison is made between the accuracy and the performance of the different variants of the adaptive finite element solver for a typical EHL problem. The test case considered in this work is given in Table 7.1. Moreover, two suitable initial coarse meshes are used. There is no specific reason in the choice of these initial meshes other than to produce a relatively good starting solution and allow the sensitivity to the choice of initial mesh to be considered. The first initial mesh is composed of a total of 16671 points where 487 of them lie on the surface common to the fluid domain. This means that this initial mesh is relatively fine close to the contact region compared to the remaining region of the elasticity domain. In the second choice of an initial mesh, relatively small mesh sizes are used yielding a mesh with 22234 points in total, of them 691 points are in the fluid region.

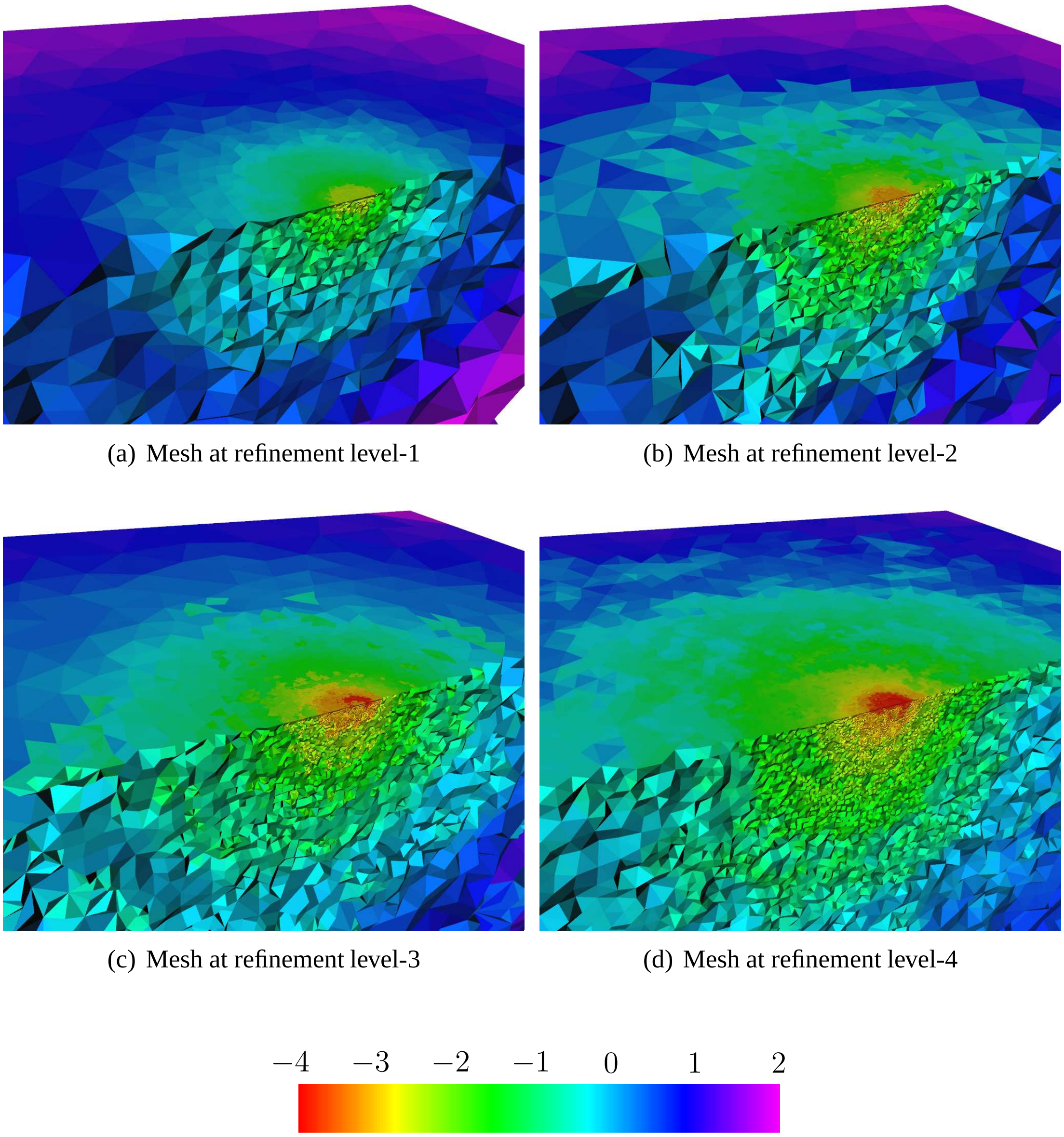
7.8.1 Implementation of the Error Estimator

Recall from previous chapters that a fully-coupled EHL problem consists of solving the Reynolds equation, the linear elasticity equation and the load balance equation simultaneously. For point contact problems, the linear elasticity equation is numerically solved on a 3D domain Ω, while the Reynolds equation is solved on a 2D fluid domain Ω_f which is a small part of the surface boundary of Ω. The solution of the linear elasticity equation exhibits large variation close to the fluid region. In other words, the mesh elements close to the fluid region show large errors. Performing local refinement in that region not only improves the accuracy of the elastic deformation solution but also increases the spatial resolution in the fluid domain for the solution of the Reynolds equation. Together, the increase in the spatial resolution and the relatively accurate elastic deformation solution yield a significantly improved pressure profile. Hence, with the improved traction boundary condition, the elastic deformation thus computed will be more accurate. Hence, a simple and an effective way to develop an adaptive procedure for a fully-couple EHL problem is to apply the local refinement to the linear elasticity mesh (with local error estimation based on the solution of the linear elasticity equation). Therefore, in this work, the error estimator discussed in Section 7.3 is only applied to the linear elasticity solution to find an approximation of local and the global errors.

Figure 7.1 shows a cut through the centreline of the 3D domain after different iterations of h-refinement based adaptivity. The elements are coloured using their element sizes. Hence the elements with very small mesh sizes ($h_e \approx e^{-4}$) are shown by red and those with large ($h_e \approx e^2$) are shown by purple. One can see that the local refinement is targeting mainly those regions close to the contact region. However, as the refinement levels go up, the refinement also extends to the regions away from the fluid region. Moreover, Figure 7.1(c) shows an arc-shaped region (corresponding to the pressure-ridge region) showing much finer elements. This explains that this is a region where the pressure-ridge affects the elastic deformation solution more significantly. Overall, this experiment suggests that the refinement strategy implemented here seems to be effective for a fully-coupled EHL problem. The corresponding 2D mesh for the Reynolds equation is getting finer in the region where it is desired to be.

7.8.2 Accuracy Appraisal

In this subsection, the accuracy appraisal of different variants of the solver is considered. As a first case, an initial mesh with 16671 mesh points is used as a base level mesh. The EHL problem is set up and solved on this starting mesh. Once the solution is obtained, local error estimation on each element of the mesh is made according to equation (7.1), while a global error estimation is obtained according to equation (7.2). Having the local error estimate for each element in hand, a set of elements are marked for refinement according to equation (7.5) where we have chosen the constant $c = 0.1, 0.2 \,\&\, 0.3$ for demonstration and comparison purposes. Note that a small value of this constant c targets significantly more elements for refinement and vice versa. As soon as the refinement is performed, the procedure is repeated again until the maximum number of levels specified are reached. Recall that variant 2 of the solver also performs an optimization process on



(a) Mesh at refinement level-1

(b) Mesh at refinement level-2

(c) Mesh at refinement level-3

(d) Mesh at refinement level-4

(e) Color scheme used for different values of $\ln$(element size).

Figure 7.1: A view of meshes at different refinement levels based upon an initial mesh with 16671 points.

the refined meshes at each refinement level while variant 3 only applies the optimization process at last refinement level.

For different mesh refinement strategies, Table 7.2 shows a comparison of behaviour of problem sizes (both in the pressure unknowns and total problem sizes) and the solution

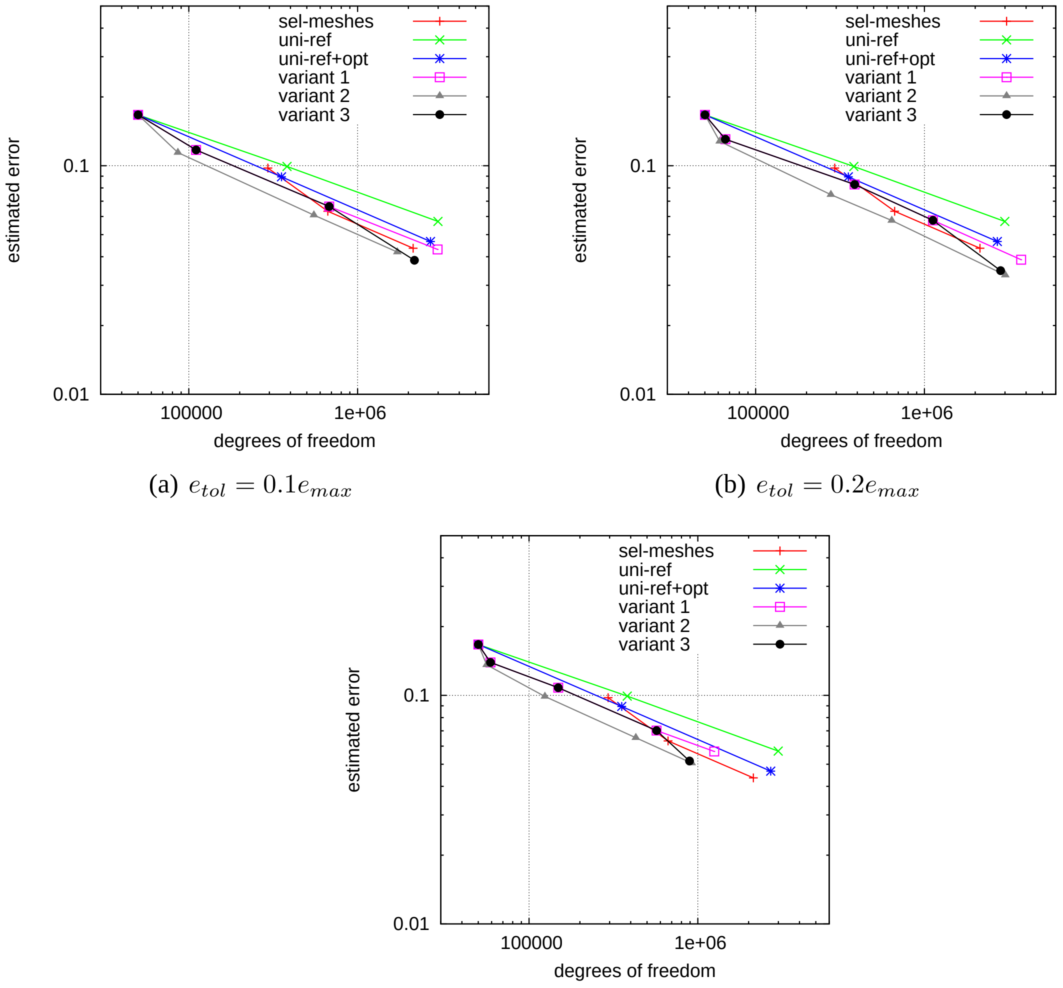
Table 7.2: Statistics for solutions using uniform refinement and adaptive h-refinement. Variant 1 performs no mesh optimization, variant 2 performs optimization at every level, and variant 3 performs optimization at the finest level only.

level	uniform refinement		h-refinement								
			variant 1			variant 2			variant 3		
	non-opt.	opt.	$c = 0.1$	$c = 0.2$	$c = 0.3$	$c = 0.1$	$c = 0.2$	$c = 0.3$	$c = 0.1$	$c = 0.2$	$c = 0.3$
number of pressure unknowns n_p											
0	431	431	431	431	431	431	431	431	431	431	431
1	1777	1777	1539	897	699	1539	897	699	1539	897	699
2	7217	7217	5163	3357	2265	5602	3489	2496	5163	3357	2265
3	-	-	13700	8679	4674	12603	7477	4576	13700	8679	4674
4	-	-	-	16874	10016	-	19231	10569	-	16874	10016
Total degrees of freedom											
0	50043	50043	50043	50043	50043	50043	50043	50043	50043	50043	50043
1	381809	354230	110422	66136	59062	86125	61210	56068	110422	66136	59062
2	2994948	2704035	679186	385831	148894	551030	279022	124183	679186	385831	148894
3	-	-	2979240	1122655	569128	1719745	639962	429131	2170815	1122655	569128
4	-	-	-	3739788	1250907	-	3011678	918850	-	2827140	892530
central film thickness H_c											
0	0.39677	0.39677	0.39677	0.39677	0.39677	0.39677	0.39677	0.39677	0.39677	0.39677	0.39677
1	0.42500	0.42446	0.42210	0.40666	0.40121	0.42215	0.40644	0.40104	0.42210	0.40666	0.40121
2	0.43071	0.43002	0.42826	0.42479	0.42398	0.42876	0.42482	0.42290	0.42826	0.42479	0.42398
3	-	-	0.42996	0.42931	0.42624	0.42929	0.42829	0.42597	0.42997	0.42931	0.42624
4	-	-	-	0.43025	0.42934	-	0.43024	0.42903	-	0.43027	0.42922
minimum film thickness H_m											
0	0.26047	0.26047	0.26047	0.26047	0.26047	0.26047	0.26047	0.26047	0.26047	0.26047	0.26047
1	0.28472	0.28442	0.28301	0.27163	0.26543	0.28318	0.27208	0.26572	0.28301	0.27163	0.26543
2	0.29051	0.29112	0.28995	0.28715	0.28427	0.29023	0.28744	0.28483	0.28995	0.28715	0.28427
3	-	-	0.29112	0.29034	0.28874	0.29067	0.28947	0.28784	0.29111	0.29034	0.28874
4	-	-	-	0.29133	0.29051	-	0.29121	0.29036	-	0.29129	0.29055

(in terms of central and minimum film thicknesses). In the case of uniform refinement (optimized and non-optimized), the pressure unknowns are increasing by about a factor of four while the total problem size by a factor of about eight. On the other hand the local refinement process targets elements for refinement showing large errors and the problem sizes behave differently for different values of c. In other words, a larger value of constant c directs the refinement process to be more specific as to refining the elements showing the largest errors. Note that in each case, the local refinement mostly affects the elements close to the contact region (see, also Figure 7.1). Moreover, it can be seen that variant 1, for $c = 0.1$, results in approximately the same solution after two levels of refinement compared to that with the uniform refinement cases. In this case, the pressure unknowns are almost the same as with the cases of uniform refinement but the total problem size is about a quarter. Increasing the value c further reduces the total problem size; however, one needs to have a little compromise on the accuracy of the solution. Variant 2, which optimizes the meshes at every refinement level, seems to yield better accuracy in results than variant 1 with relatively smaller problem sizes. Note that if it would be possible to perform a third level of uniform refinement (with or without optimization) then this would lead to a very large problem size (total problem size would increase by about a factor of eight). In such a case, it would be easier to see the significance of local refinement process.

It should be noted that the output of variant 3 differs from variant 1 only at the finest level. Here, optimization of refined meshes leads to a significant decrease in the total size of the finest level problems while ensuring the overall accuracy of the solution. In other words, for each choice of c, variant 3 yields the same accuracy in the solution (compared to variant 1) with a smaller problem size at the finest level. It should be noted that the optimization of meshes in variant 3 at the final level ensures the accuracy of the pressure profile (see Chapter 6). Thus, if the pressure profile is more accurate then the film thicknesses solution will be more reliable. On the other hand variant 2 optimizes the meshes at every refinement level therefore it ends up with a smaller problem size at the finest level compared to variant 3. Although, the results are not fully converged one can see that the difference in the output of variants 2 & 3 gets increasingly small for different choices of c (especially for $c = 0.1$ & 0.2), and appears to converge to the same solution. Finally, it can be seen that, for $c = 0.1$ & 0.2, the central film thicknesses solution appears to converge more quickly as the refinement level goes up. However, the situation is slightly different for the minimum thicknesses solution. The reason may be that the local refinement process mainly targets the pressure ridge region (see Figure 7.1) and hence the corresponding changes in the pressure ridge might influence the minimum film thicknesses solution. In short, results show that both variants 2 & 3 end up with the same accuracy in their solution with relatively small problem sizes compared to variant 1. Furthermore, we shall see next that both variants 2 & 3 result in better accuracy than variant 1 and the uniform refinement cases quite effectively.

For each choice of c, a comparison of the estimated global errors obtained for each variant of the solver is shown in Figure 7.2. Note that the global error estimation is for the elasticity solution with a converging pressure profile (different for each mesh strategy) as the traction boundary condition. The case of uniform refinement (with and without optimization) along with the selected mesh cases (produced and considered in the previous chapter) are also included. One can see that a non-optimized uniform refinement process leads to small reduction in the error with increasing problem size. But, if the meshes are optimized after each uniform refinement process then a relatively fast reduction in the error is obtained. In this example, for each refinement criterion, the local refinement cases (all three variants) appeared to have a superior error reduction rate, with respect to problem size, as compared to both cases of uniform refinement. It can be seen that optimization of meshes at each refinement level further improves the rate of error reduction with respect to the problem size. It should also be noted that the last level optimization (variant 3) significantly reduces the error at the finest level, and results in approximately



(a) $e_{tol} = 0.1e_{max}$

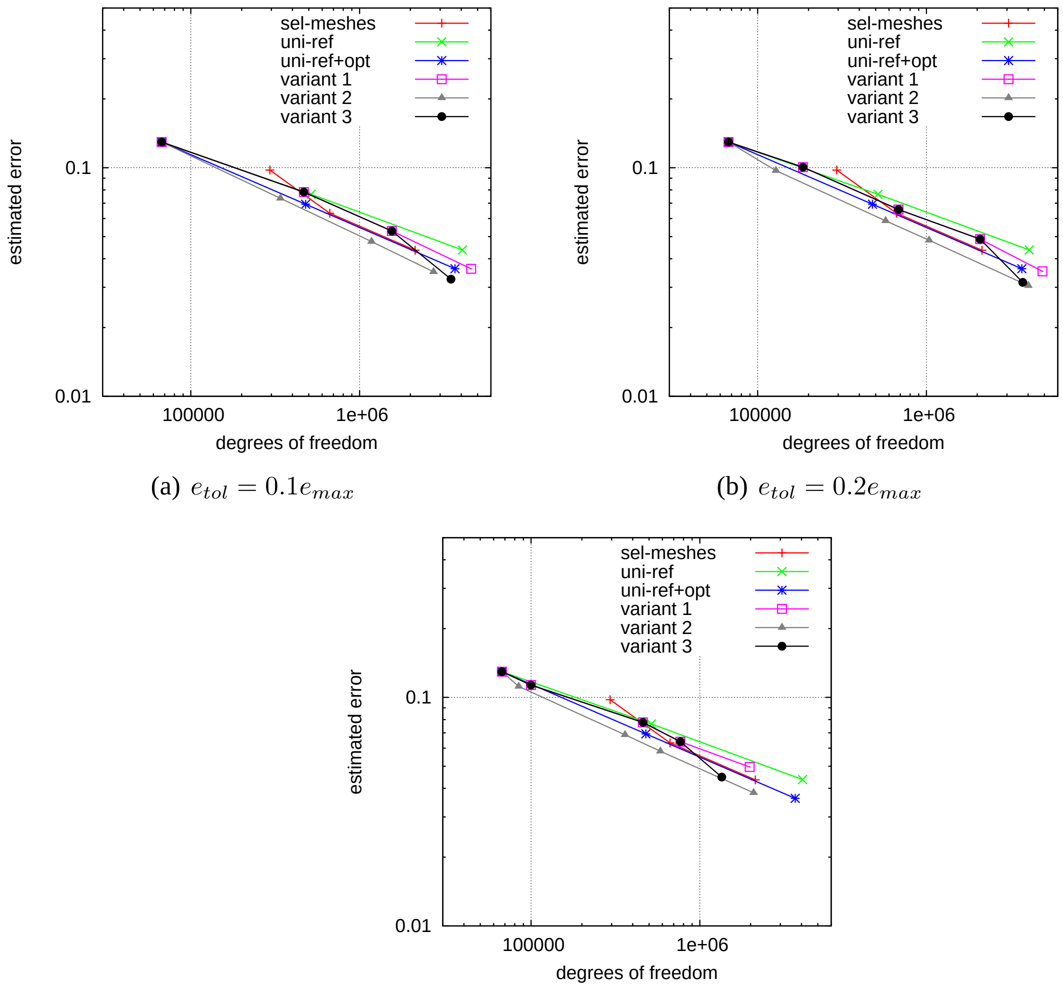
(b) $e_{tol} = 0.2e_{max}$

(c) $e_{tol} = 0.3e_{max}$

Figure 7.2: A comparison of global error estimation for different mesh refinement strategies using a coarsest mesh of 16671 points.

the same accuracy as that obtained with the optimization at every level (variant 2). Finally, the selected mesh cases even perform better than the local refinement without post-optimization of meshes (variant 1). Furthermore, it can be seen that different refinement criteria used here only controls the problem sizes, and does not affect the error's convergence rate with a problem size. In other words, the adaptive technique is not too sensitive to the choice of the parameter c.

As a second test case, a different initial mesh composed of 22234 mesh points is considered. This initial mesh is relatively more fine than that used above. Figure 7.3 shows the accuracy appraisal for different variants of the solver compared to the use of uniform refinement and the selected mesh cases. The same behaviour in the results can



(a) $e_{tol} = 0.1e_{max}$

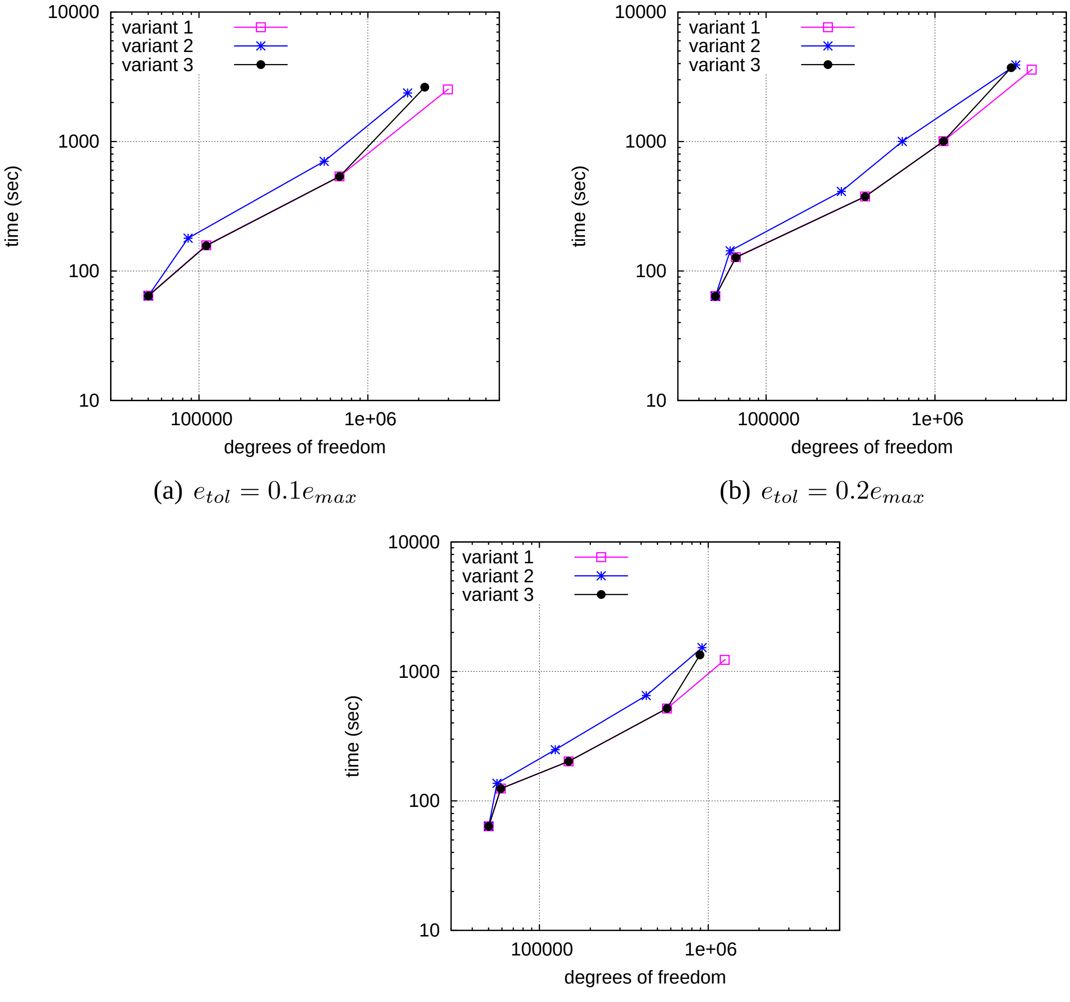
(b) $e_{tol} = 0.2e_{max}$

(c) $e_{tol} = 0.3e_{max}$

Figure 7.3: A comparison of global error estimation for different mesh refinement strategies using a coarsest mesh of 22234 points.

be observed as before however, the case of optimized uniform refinement shows a better error reduction rate compared to the non-optimized local refinement case. Nevertheless, it can be seen that, again, the local refinement cases (both with optimization at only the last or at every level, variants 3 & 2 respectively) perform better than the other cases in terms of accuracy.

As a whole, one can conclude from these experiments that the local refinement of meshes with post optimization at only the final or at all levels results in more accurate results with a relatively small problem size. Most importantly, the adaptive algorithm (with at least last level optimization) leads to better results compared to the selected mesh cases. In this sense, the use of automatic mesh refinement based upon ‘a posteriori’



(a) $e_{tol} = 0.1e_{max}$

(b) $e_{tol} = 0.2e_{max}$

(c) $e_{tol} = 0.3e_{max}$

Figure 7.4: A comparison of performance of different variants of adaptive finite element solver using the coarser initial mesh.

error estimation has produced better meshes than the hand-tuning approach described in Chapter 6. In the next subsection, the performance of each of the variants of the proposed adaptive solver is assessed for each of the three refinement criteria.

7.8.3 Performance

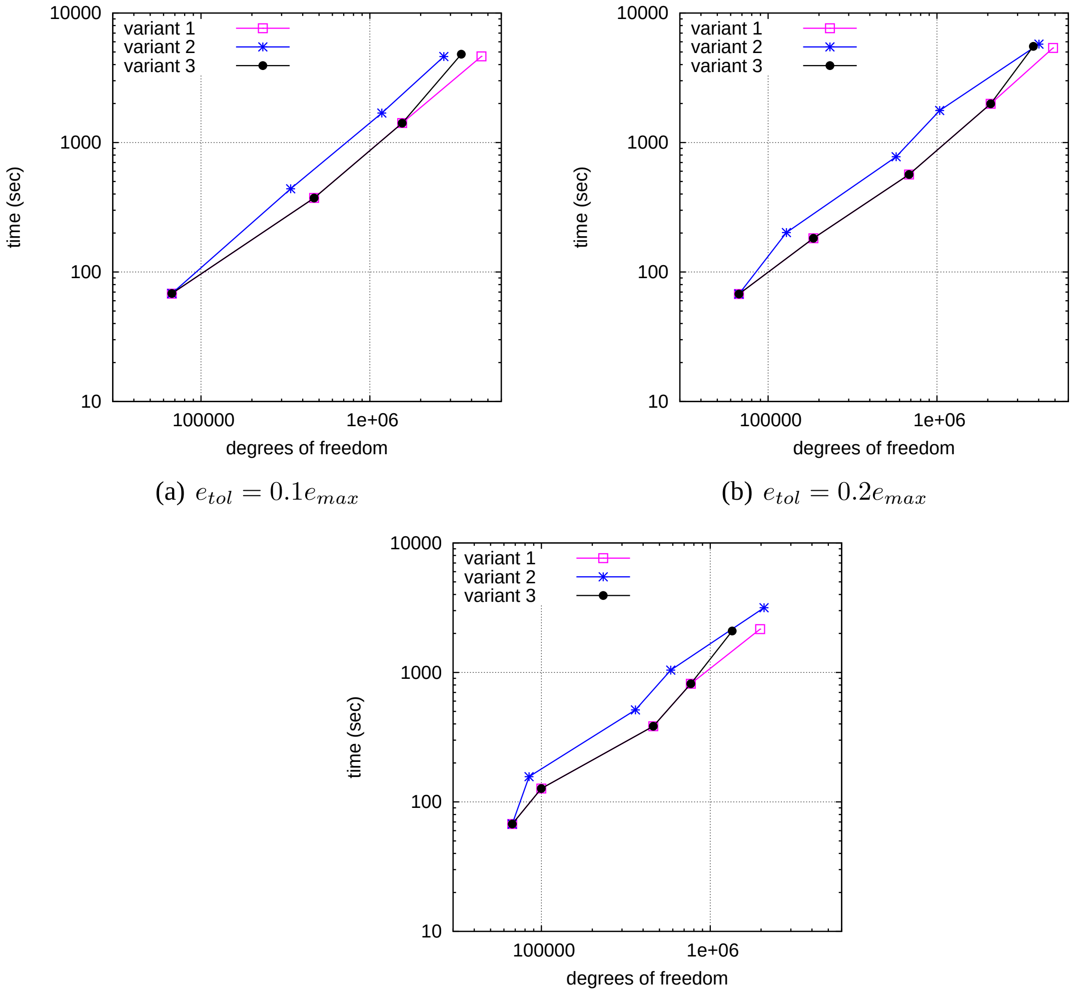
In this subsection, the performance of different variants of adaptive finite element solver are assessed. For the initial mesh case 1, the computational times are plotted in Figure 7.4 for each of the three adaptive refinement criteria. In Figure 7.4, a jump in the growth of computational time can be observed while switching from base level to first level.

Table 7.3: Statistics of solution at different refinement levels. Variant 1 performs no optimization, variant 2 perform optimization at every level, and variant 3 performs optimization at the finest level only.

level	uniform refinement		h-refinement								
			variant 1			variant 2			variant 3		
	non-opt.	opt.	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
Total nonlinear iterations											
0	14	14	14	14	14	14	14	14	14	14	14
1	9	9	8	10	11	9	9	9	8	10	11
2	4	4	4	5	4	4	4	4	4	5	4
3	-	-	5	4	4	3	3	3	3	4	4
4	-	-	-	5	4	-	3	3	-	3	3
Average number of linear iterations per one nonlinear iteration											
0	11.4	11.4	11.4	11.4	11.4	11.4	11.4	11.4	11.4	11.4	11.4
1	12	11	12.8	11.8	11.7	11.1	11.1	10.8	12.8	11.8	11.7
2	13.3	13	15.0	15.0	15.0	11.5	13.5	10.5	15.0	15.0	15.0
3	-	-	13.0	15.0	15.0	12.3	12.7	11.7	12.7	15.0	15.0
4	-	-	-	15.0	15.0	-	11.0	10.3	-	11.3	12.7

The reason is that the first refinement process led to refinement of only a few elements leaving the problem size approximately unchanged. In other words, the computational time is almost doubled for almost the same sized problem at the first refinement level (see Section 7.8.5 for further details). Moreover, the variant 2 applies an optimization process on the refined mesh which leads to a slightly smaller problem size but the total time has increased compared to other two variants. After the first level, the growth in the time appeared to be almost linear for each of the variants however, variant 3 shows a jump in the computational time on the final level which is due to optimization process on the last level mesh. Furthermore, for each refinement criterion, all three variants appeared to be computationally similar despite different problem sizes. In other words, the optimization of the refined meshes, at least at the final level, leads to little change in computational time (but to relatively more accurate results, as discussed above).

Table 7.3 gives statistics of average number of linear iterations and the number of nonlinear iterations for each variant of the adaptive solver using the three different refinement criteria. It can be seen that as the refinement level increases in each case, fewer nonlinear iterations are required to achieve convergence. Importantly, the performance of the solver seems independent of the adaptivity method used. Moreover, the optimization of meshes at final level in variant 3 results in a relatively small number of nonlinear iterations compared to variant 1. Similarly, variant 2 requires even fewer nonlinear iterations at the intermediate levels as well compared to other two variants of the solver. In addition to nonlinear iterations, variant 3 requires slightly less average number of linear iterations per nonlinear iterations at the final level while, on the other hand, this number is also reduced for variant 2 at the intermediate levels as well. *The most important observation of all*



(a) $e_{tol} = 0.1e_{max}$

(b) $e_{tol} = 0.2e_{max}$

(c) $e_{tol} = 0.3e_{max}$

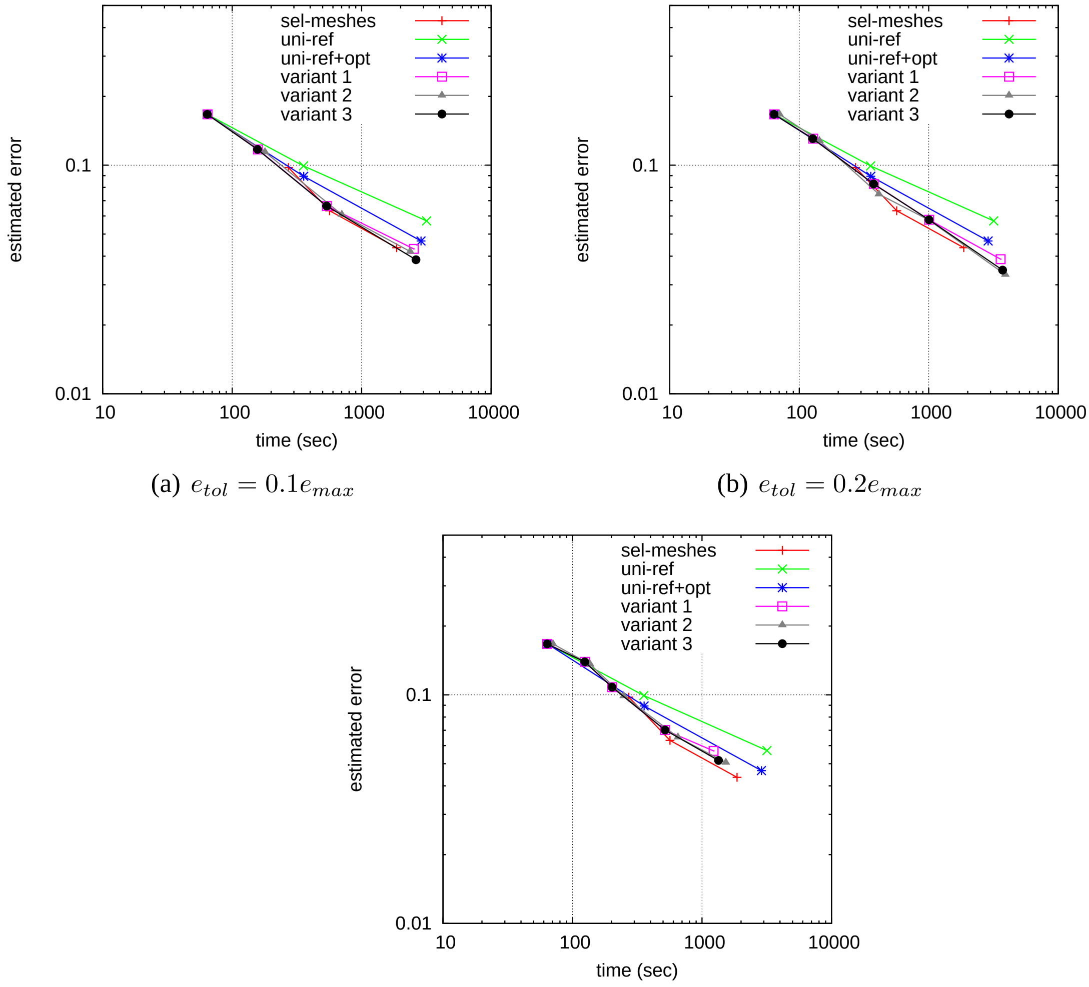
Figure 7.5: A comparison of performance of different variants of adaptive finite element solver using the finer initial mesh.

however is that overall, the average number of linear iterations per nonlinear iteration appears to be independent of the problem sizes for each variant of the solver.

As a next case, Figure 7.5 shows a similar behaviour in the computational times while starting with initial mesh case 2. No jump in the growth of time is observed on the first level for $c = 0.1, 0.2$ as these values lead to refinement of a lot more elements compared to the case for $c = 0.3$. A similar deterioration in time can be observed at any other refinement level provided the original problem size is not significantly altered as a result of local refinement process. Finally, ignoring the additional time of optimization of meshes (which also appears to be linear), all three variants of the solver appear to be close to optimal, with approximately linear growth in the computational time. The qualitative behaviour of the iteration counts is similar to that shown in Table 7.3.

7.8.4 Further Discussion

In this subsection, an overall comparison between the behaviour and efficiency of different schemes is presented. Note that all cases presented here make use of AMG preconditioning of the elasticity block, as the use of GMG preconditioning is not possible while taking into account the optimization of meshes within an adaptive algorithm. Figure 7.6 shows a comparison of the estimated global error with respect to the computational time for different schemes using the initial mesh case 1. The selected mesh cases (see Chapter 6) are also included to make it an overall comparison. It can be seen that the selected mesh



(a) $e_{tol} = 0.1e_{max}$

(b) $e_{tol} = 0.2e_{max}$

(c) $e_{tol} = 0.3e_{max}$

Figure 7.6: A comparison of performance of different solvers.

cases and the different variants of the adaptive algorithm are efficient in reducing the error compared to uniform refinement cases. Moreover, for $c = 0.2$ & 0.3, the selected mesh cases appear to be efficient compared to different variants of the adaptive algorithm; however, this difference is very small. Note that each variant of adaptive algorithm is fully automatic in optimizing the computational process. On the other hand, the meshes used in the selected mesh cases are based on a large number of experiments to obtain a desired accuracy at minimal cost. As a whole, the adaptive algorithm appears to be more convenient than the selected mesh cases despite a slight increase in the computational cost (for $c = 0.2$ & 0.3). Furthermore, both the variants 2 & 3 of adaptive algorithm are comparatively better than the variant 1 in reducing the overall error at a fixed computational cost.

7.8.5 Accuracy of Intermediate Solves

The results presented so far were obtained by solving the nonlinear EHL problem to full accuracy at each refinement level. However, it is generally not necessary to solve the problem too accurately at each intermediate level. In other words, it is only necessary to solve a problem to a sufficient precision to obtain a good approximation to the solution in order to direct the adaptive procedure. In this subsection, the effect of different stopping tolerances for nonlinear solves at each of the intermediate levels is discussed. It should be noted that the final level problem will always be solved accurately. For this purpose, an experiment is setup using variant 3. Recall that variant 3 only performs optimization on the refined meshes at the final level. In this experiment, refinement criterion: $e_{tol} = 0.25\, e_{max}$ is used. Note that, there is no specific reason in the choice of variant 3 of the solver and the refinement criterion other than to make it a typical test. A total of four refinement levels are used in this experiment, with initial mesh case 1 as a base level mesh. The results obtained for different stopping tolerances for the Newton solver are given in Table 7.4, in terms of the number of pressure unknowns (np), the total problem size, the nonlinear iterations (ni), the linear iterations (li) and the total solve time (excluding time for optimization at final level), the optimization time at the final level and the global error estimation.

Note that significant savings in the computational times are achieved with an increase in the tolerance. The use of tolerance as high as 10^{-1} leads to about 25% savings in the total solve time while keeping the other values almost unchanged. A further increase in the tolerance to 10^{-0} affects the refinement process slightly. This tolerance results in a slightly smaller problem with a relatively large error. Most probably, the quality of initial

Table 7.4: Effect of different stopping tolerances for intermediate level nonlinear solves upon the overall performance of the adaptive solver.

tol	np	total dof	ni	li	time (sec)	opt-time (sec)	estimated global error
$U^{\frac{1}{3}}$	12835	1569053	3	36	1640	669	0.0429396
10^{-3}	12818	1567527	3	36	1495	666	0.0429393
10^{-2}	12768	1562842	3	34	1276	666	0.0429871
10^{-1}	12747	1564912	3	32	1262	669	0.0429649
10^{-0}	12323	1277703	5	61	1327	537	0.0461213

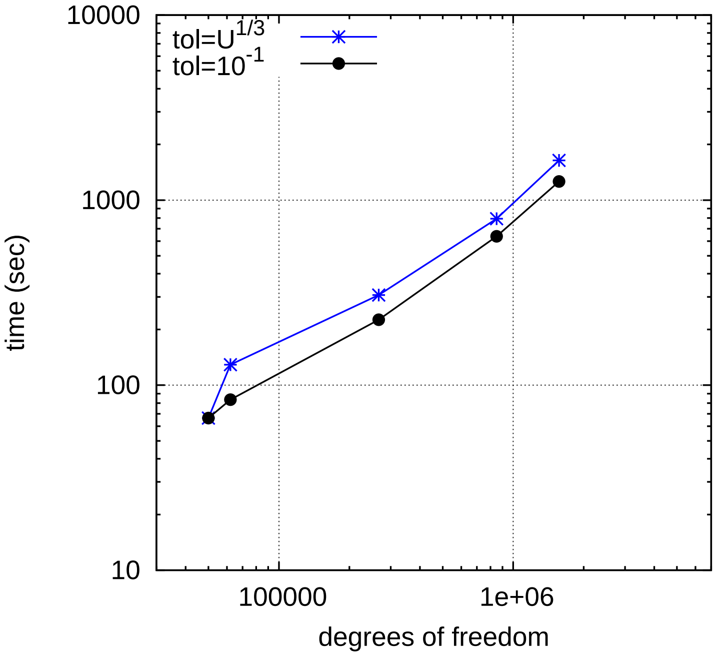


Figure 7.7: The effect of tolerance for the intermediate solves over the performance of an adaptive procedure.

guess is also not so good causing the computational work to slightly grow. Hence, an intermediate tolerance of 10^{-1} is recommended on the basis of this test.

Finally, Figure 7.7 shows the behaviour in the growth of the computational time for the accurate and approximate solves at intermediate levels (excluding the optimization time). One can see that the jump in the computational time at the first level has not appeared in the case of the approximate solve, and the algorithm has led to a smooth linear growth in the computational time. Note that for each level i ($i = 1, 2, 3, 4$), the problem is solved approximately until the $(i-1)^{\text{th}}$ level.

7.8.6 A Modified Error Estimator

The results presented so far were obtained using the ZZ-error estimator discussed in Section 7.3. For the EHL problems, it is even possible to simplify this error estimator further. Recall that a fully-coupled EHL point contact approach (used in this work) involves a numerical solution of the linear elasticity equation on a 3D domain. There are three solution components corresponding to each mesh point, which represent displacement components in each spatial direction. Amongst them the most dominant components of the solution vector are those in the z-direction, see Section 2.6. Hence, such solution components can exhibit relatively large errors on coarse grids due to the large variation in their values. Therefore, it may be quite useful to only consider those solution components to find an approximation of an error. Let w_h represent a finite element approximation of the z-displacement solution component, which can be written as:

$$w_h = \sum_{i=1}^{n} N_i w_i \ ,$$

where N_i represents the basis function corresponding to a node i, and n is total number of nodes. The gradient of the above approximation can be written as:

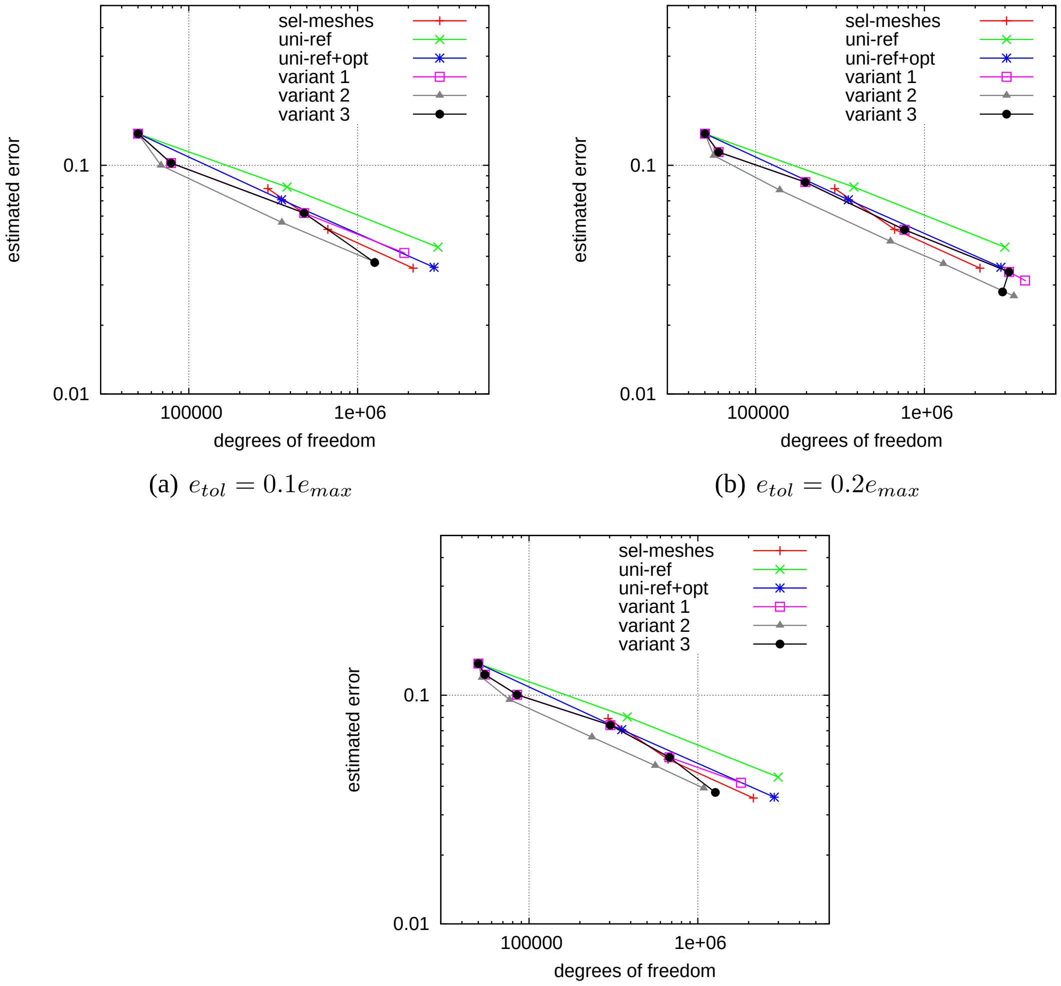
$$\nabla w_h = \sum_{i=1}^{n} \nabla N_i w_i \ . \tag{7.6}$$

Thus replacing the stress vector in equation (7.1) with the expression given in equation (7.6), a simple error estimator can be defined in the L_2 norm as:

$$\|\mathbf{e}^*_{\nabla w}\|^2 = \int_\Omega (\nabla w^* - \nabla w_h)^T (\nabla w^* - \nabla w_h) \mathrm{d}\Omega \ , \tag{7.7}$$

where the recovered gradient ∇w^* is obtained using the same procedure as described in Section 7.3 (i.e. taking a piecewise linear approximation based upon recovered nodal values obtained by averaging over elements surrounding each node). Some provisional results using this simple error estimator to assess its applicability are now presented.

Figure 7.8 shows a comparison of the error estimates for different variants of the solver together with the selected mesh cases and the uniform refinement cases (both with and without optimization). Starting with initial mesh case 1, for each variant of the solver, a total of three refinement levels have been used for a refinement criterion with $c = 0.1$ and a total of five refinement level in the other two cases. One can see a similar behaviour in the results as was observed in the previous subsection. For each of the refinement



(a) $e_{tol} = 0.1 e_{max}$

(b) $e_{tol} = 0.2 e_{max}$

(c) $e_{tol} = 0.3 e_{max}$

Figure 7.8: A comparison of global error estimation for different mesh refinement strategies using a coarsest mesh of 16671 points.

criteria, variant 1 appears to result in an equivalent accuracy compared to the optimized case of uniform refinement. However, the selected mesh cases are even more efficient. On the other hand, variant 2 shows a better reduction rate of error (with respect to a problem size) throughout however, variant 3 tends to yield almost same accuracy at the finest level.

Finally, we note that this simplified version of the original estimator is relatively easy to implement and seems to work just as well as the original estimator. However, it will be necessary to undertake a number of further experiments for different base level meshes and the different EHL cases to demonstrate its applicability in full.

7.9 Conclusion

In this chapter, an adaptive finite element solution to a fully-coupled EHL problem has been discussed. A ZZ-error estimator has been used to find the local error approximations and these local error estimations are used to find an estimation of the global energy error. These error estimations have been used to mark elements for refinement which were exhibiting larger errors than a prescribed tolerance. The refinement criterion used at each refinement level was based on the largest of the local error estimates. The local refinement of the meshes was carried out using the algorithm that is described in Section 7.5. Three variants of the adaptive algorithm are considered in this work. The first variant applies a standard h-adaptive algorithm. The second variant considered the post-optimization of the meshes at each refinement level in order to increase the accuracy. With the post-optimization process for the meshes, a new mesh was obtained at each level which means that the hierarchy of meshes does not exist anymore. Thus, neither the derefinement nor the use of GMG based preconditioner is possible. Variant 3 of the adaptive solver only utilized the optimization at the final level in order to avoid the possibility of excessive green elements on the 2D surface mesh (which remains unchanged by the optimization process).

The accuracy appraisal of all three variants of the solver were made using two different initial meshes against the use of uniformly refined meshes (both optimized and non optimized) and against the efficient meshes selected in the previous chapter. The results showed that both the variant 2 and the variant 3 appeared to perform best in terms of accuracy. In other words, variant 2 & 3 have close resemblance with an hr-adaptive algorithm (at least at the final level) resulting in better results. A drawback of the optimization of the meshes was to lose the hierarchy of meshes and the data. However, unchanged surface meshes allowed us to generate a better initial guess (by solving a linear elasticity problem with the interpolated boundary condition) to reduce the computational work at the subsequent levels. Moreover, all three variants of the solver showed almost a linear growth in their computational time despite of quite few zig-zag behaviours.

Significantly, it was shown that an approximate solve at each of the intermediate levels leads to a smooth linear growth in the computational time. Furthermore, due to additional time required for the optimization process of meshes, variants 2 & 3 require a slightly longer time than the variant 1 (for a fixed problem size). However, this slight compromise on the computational work ensures that these variants result in a more accurate solution compared to variant 1. Finally, a simplified version of the ZZ-error estimator was implemented. This estimator is only based on computing the gradients of the z-components of the solution of the elasticity problem, and is relatively easy to implement. The initial results showed that the modified estimator can also be effective in an adaptive fully-coupled EHL finite element point contact procedure.

Overall, it is demonstrated that automatic control of the local mesh adaptivity results in better meshes than were obtained by experimentation in Chapter 6, and that the optimal convergence behaviour of the preconditioned iterative solver introduced in Chapter 4 & 6 is still maintained within this adaptive algorithm. Furthermore, only moderate accuracy is required from the iterative solves at the intermediate mesh levels in order to guide the local refinement procedure effectively.

Chapter 8

Conclusion

8.1 Summary

In this research a numerical study into the efficient solution of fully-coupled EHL line and point contact problems has been undertaken. This is an extension of previous work of Habchi [49–51] who first used a coupled approach to solve EHL problems based upon the elastic deflection in the film thickness equation being modelled by using a finite element solution of Lamé's equation of linear elasticity. Contrary to the traditional half-space approach for elastic deflection the finite element model only uses information at the neighbouring points to calculate the elastic deflection at a point in the domain. This results in a highly sparse matrix and makes it easier to use sparse matrix methods. Furthermore, the strong coupling makes it possible to reach the solution without any special treatment for convergence. The drawback, however, is the need to solve the elasticity equation in a 2D domain for line contact problems and a 3D domain for point contact problems, which makes the size of computational problem very large.

In this thesis the issue of the high computational and memory costs of the fully-coupled approach to solve EHL problems was discussed. Habchi et.al. [49–51] used a sparse direct solver to solve the linearized system at each Newton step. To solve very large sparse problems, iterative methods are considered to be superior to sparse direct methods in terms of memory, however good preconditioners are required to make them computationally competitive. Therefore the main objective of this study has been to develop a fast and efficient preconditioned Newton-Krylov solver for fully-coupled EHL problems. In particular, a new blockwise preconditioner has been developed which is designed to exploit the specific structure of this problem and combines algebraic multigrid (AMG) /geometric multigrid (GMG) for the linear elasticity block with a separate, efficient, approximation to precondition the Reynolds part.

In Chapter 4, the accuracy of our implementation of both the line contact solver and the point contact solver have been validated against published results in the literature. In Chapter 5, the accuracy and performance of different variants of the proposed line contact solver was assessed. It was shown that the accuracy of the computed EHL solution is not significantly affected using a suitable set of selected non-uniform meshes. Moreover, it has been shown that the AMG preconditioned variant of the iterative solver performs better than the sparse direct solver. The GMG preconditioned variant of the iterative solver was observed to be even more efficient than the AMG preconditioned variant, however one needs to accept restrictions over the meshes due to the global refinement needed to generate the hierarchy of meshes for GMG. Later on, the effect of varying the Poisson ratio (for the 2D elastic model) over the accuracy and the performance of line contact solvers was demonstrated. It was shown that the accuracy of the EHL solution is not affected with the variation of Poisson ratio provided a sufficiently fine elasticity mesh is used. Moreover, it was demonstrated that the performance of preconditioned iterative solver remains optimal as the Poisson ratio is increased. (However, as the limit of 0.5 is approached the preconditioned iterative solver finally deteriorates.)

In Chapter 6, the accuracy and the performance of the proposed the EHL point contact solver is presented. First of all, a large number of experiments have been carried out to investigate the accuracy of the elastic deformation as well as the EHL solution over a number of non-uniform 3D meshes. It was possible to select a set of efficient meshes such that a substantial reduction in the total computational cost was achieved without significantly affecting the accuracy of the EHL solution. Moreover, the application of the preconditioning strategy was shown to significantly outperform the sparse direct solver, with huge savings in memory and time being obtained. As with the line contact case, a further saving in the time and the memory growth is obtained using the GMG preconditioning of the elasticity block. However, these savings are not sufficient to overcome the restriction from using a global refinement process to generate the hierarchy of meshes. Furthermore, it was shown that both the time and memory growth appeared to be linear with increasing problem sizes. Finally, it was shown that unstructured hierarchical meshes without appropriate mesh optimization can lead to poor quality EHL results: again suggesting a preference for the use of AMG over GMG.

In Chapter 7, a locally adaptive finite element procedure for fully-coupled EHL point contact problems was discussed. The local h-refinement was based on the approximation of the local error in the energy norm using the ZZ-error estimator. The main adaptive solver was distinguished into different variants on the basis of post-optimization (or not) of the locally refined meshes. It was shown that the adaptive procedure with at least the last level of the locally refined meshes being optimized performs better in terms of accuracy. Moreover, it was shown that the application of the preconditioning strategy also performs well for this locally adaptive finite element scheme, and that a linear growth in the computational cost is still observed. Finally, some initial results were presented for a simplified version of the ZZ-error estimator which was based on the gradients of the z-component of the solution of the elasticity problem. These provisional results showed that the modified estimator appears to work as well as the original estimator.

8.2 Future Work

In this thesis an efficient preconditioned iterative solution to fully coupled EHL line and point contact problems has been discussed. Numerical results show the effectiveness of the strategy used both for line and point contact problems: importantly the number of linear iterations at each nonlinear solve is independent of the size of the problem, and both the computational times and the memory growth are almost optimal with increasing size problems. Especially for the fully coupled point contact problem, huge savings in the computational times and the memory growth are made possible. Furthermore, the proposed strategy has also been shown to be effective as part of a locally adaptive finite element solution of the point contact problem. Nevertheless, there are still possible ways to improve, and further extend, this work to obtain even better results for a variety of EHL problems.

- Firstly, it must be acknowledged that due to the time constraints associated with this project it has only been possible to test the proposed techniques on a modest number of problems with a selection of parameters. In order to fully test the robustness of our techniques it will be beneficial to consider even more loading cases, a wider selection of coarse grids, and a broader range of lubricants models.

- In Section 4.4, the Reynolds block J_{11} was used effectively as an approximation of the Schur complement $S = J_{11} - J_{12}J_{22}^{-1}J_{21}$, and an efficient sparse direct approach was used to calculate the effect of its inverse over an arbitrary vector (i.e. $z_p = J_{11}^{-1}r_p$). However, it may be even possible to consider the solution $Sz_p = r_p$ by means of an iterative method with J_{11} as a preconditioner for it. Moreover, as the J_{11} is the discretization of an advection diffusion block, so the use of AMG preconditioning may also be effective.

- A natural extension of this work is to parallelize the EHL code using the MPI library [42, 45, 47]. This is possible with the use of parallel sparse direct solver [5] and the use of a parallel AMG approach [79].
- Although the preconditioning strategy developed in this work to solve the fully-coupled EHL problem is shown to be highly effective, this is only using linear finite elements. An important extension will be to make use of higher order elements, as shown by Habchi et al. [49–51], with the development of an efficient AMG technique for the elasticity problem using higher order elements (extending [106] for example). It is to be expected that this would allow even better performance results especially for point contact problems, where it was not possible with the use of linear elements.
- It is not necessary to restrict the above extension to continuous higher order elements. For example, Lu et al. have shown that higher order discontinuous Galerkin (DG) finite elements may be used effectively to discretize the Reynolds equation [70–72].
- With the extension of this work to make use of both parallelism and higher order elements, the extension to transient EHL point contact problems [45, 46], may well become feasible without excessive run times. This would be a very significant development.

Bibliography

[1] HSL, a collection of Fortran codes for large-scale scientific computation,See http://www.hsl.rl.ac.uk/, March, 2011.

[2] S. Ahmed, C. E. Goodyer, and P. K. Jimack. An efficient preconditioned iterative solution of fully-coupled elastohydrodynamic lubrication problems. *Applied Numerical Mathematics*, 62(5):649–663, 2012.

[3] M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*. John Wiley & Sons, Inc., New York, 2000.

[4] M. Ainsworth, J. Z. Zhu, A. W. Craig, and O. C. Zienkiewicz. Analysis of the Zienkiewicz-Zhu a-posteriori error estimator in the finite element method. *International Journal for Numerical Methods in Engineering*, 28:2161–2174, 1989.

[5] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling. *SIAM Journal of Matrix Analysis and Applications*, 23(1):15–41, 2001.

[6] T. Apel, S. Grosman, P. K. Jimack, and A. Meyer. A New Methodology for Anisotropic Mesh Refinement Based Upon Error Gradients. *Applied Numerical Mathematics*, 50:329–341, 2004.

[7] T. Apel, A. M. Sändig, and J. R. Whiteman. Graded Mesh Refinement and Error Estimates for Finite Element Solutions of Elliptic Boundary Value Problems in Non-smooth Domains. *Mathematical Methods in the Applied Sciences*, 19(1):63–85, 1996.

[8] I. Babuska, T. Strouboulis, C. S. Upadhyay, S. K. Gangaraj, and K. Copps. Validation of a posteriori error estimators by numerical approach. *International Journal for Numerical Methods in Engineering*, 37:1073–1123, 1994.

[9] M. J. Baines, M. E. Hubbard, and P. K. Jimack. A Moving Mesh Finite Element Algorithm for the Adaptive Solution of Time-Dependent Partial Differential Equations with Moving Boundaries. *Applied Numerical Mathematics*, 54:450–469, 2005.

[10] S. Bair. The Pressure-Viscosity Coefficient of a Perfluorinated Polyether Over a Wide Temperature Range. *ASME J. Tribol.*, 123(1):50–53, 2001.

[11] C. Barus. Isothermal, Isopiestics and Isometrics Relative to Viscosity. *Am. J. Sci*, 25:87–96, 1893.

[12] C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175:311–341, 1999.

[13] M. Benzi. Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics*, 182:418–477, 2002.

[14] M. Berzins. Solution-based mesh quality for triangular and tetrahedral meshes. In *Proceedings of the Sixth International Meshing Roundtable*, pages 427–436. Sandia National Laboratories, 1997.

[15] M. Berzins. Mesh quality - geometry, error estimates or both? In *Proceedings of the Seventh International Meshing Roundtable*, pages 229–237. Sandia National Laboratories, 1998.

[16] R. Blaheta. Displacement decomposition-incomplete factorization preconditioning techniques for linear elasticity problems. *Numerical Linear Algebra with Applications*, 1(2):107–128, 1994.

[17] J. Boyle, M. Mihajlovic, and J. Scott. HSL_MI20: An efficient AMG preconditioner for finite element problems in 3d. *International Journal for Numerical Methods in Engineering*, 82(1):64–98, 2010.

[18] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Elasticity Theory*. Cambridge University Press, 2007.

[19] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.

[20] A. Brandt and A. A. Lubrecht. Multilevel Matrix Multiplication and Fast Solution of Integral Equations. *Journal of Computational Physics*, 90(2):348–370, 1990.

[21] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, 2000.

[22] A. N. Brooks and T. J. R. Hughes. Streamline-Upwind/Petrov-Galerkin Formulations for Convective Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[23] P. N. Brown and Y. Saad. Hybrid Krylov Methods for Nonlinear Systems of Equations. *SIAM Journal on Scientific and Statistical Computing*, 11:450–481, 1990.

[24] R. L. Burden and J. D. Faires. *Numerical Analysis*. Thomson Brooks/Cole, London, 2005.

[25] K. Chen. *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, Cambridge, 2005.

[26] T. A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

[27] T. A. Davis. A Column Pre-Ordering Strategy for the Unsymmetric-Pattern Multifrontal Method. *ACM Transactions on Mathematical Software*, 30(2):165–195, June 2004.

[28] T. A. Davis. Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, June 2004.

[29] T. A. Davis and I. S. Duff. An Unsymmetric-Pattern Multifrontal Method for Sparse LU Factorization. *SIAM Journal on Matrix Analysis and Applications*, 18(1):140–158, 1997.

[30] T. A. Davis and I. S. Duff. A Combined Unifrontal/Multifrontal Method for Unsymmetric Sparse Matrices. *ACM Transactions on Mathematical Software*, 25(1):1–20, 1999.

[31] L. Demkowicz, W. Rachowicz, and Ph. Devloo. A Fully Automatic hp-Adaptivity. *Journal of Scientific Computing*, 17(1-4):117–142, 2002.

[32] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst. *Numerical Linear Algebra for High-Performance Computers*. Society for Industrial and Applied Mathematics, Philadephia, PA, 1998.

[33] D. Dowson and G. R. Higginson. *Elasto-hydrodynamic Lubrication*. Pergamon Press, Oxford, 1966.

[34] S. C. Eisenstat and H. F. Walker. Choosing the Forcing Terms in an Inexact Newton Method. *SIAM Journal on Scientific Computing*, 17:16–32, 1996.

[35] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*. Oxford University Press, Oxford, 2005.

[36] H. G. Elrod. A Cavitation Algorithm. *Journal of Lubrication Technology-Transactions of the ASME*, 103:350–354, 1981.

[37] A. M. Ertel. Hydrodynamic Lubrication Based on New Principles. *Akad. Nauk. SSSR Prikadnaya Mathematica i Mekhanika*, 3(2):41–52, 1939.

[38] H. P. Evans and T. G. Hughes. Evaluation of Deflection in Semi-Infinite Bodies by a Differential Method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering*, 214:563–584, 2000.

[39] H. P. Evans and R. W. Snidle. Inverse solution of the Reynolds equation of lubrication under point-contact elastohydrodynamic conditions. *ASME Journal of Tribology*, 103:539–546, 1981.

[40] J. Fish and T. Belytschko. *A First Course in Finite Elements*. John Wiley & Sons, 2007.

[41] L. Floberg. *Cavitation in Lubricating Oil Films*. Elsevier, Amsterdam, 1964.

[42] Message Passing Interface Forum. MPI: A Message-Passing Interface Standard. *International Journal of Supercomputer Applications*, 8(No.3/4), 1994.

[43] R. W. Freund. A Transpose-Free Quasi-Minimal Residual Algorithm for Non-Hermitian Linear Systems. *SIAM J. on Scientific Computing*, 14(2):470–482, 1993.

[44] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Edition)*. The Johns Hopkins University Press, Baltimore and London, 1996.

[45] C. E. Goodyer. *Adaptive Numerical Methods for Elastohydrodynamic Lubrication*. PhD thesis, University of Leeds, Leeds, UK, 2001.

[46] C. E. Goodyer and M. Berzins. Adaptive timestepping for elastohydrodynamic lubrication solvers. *SIAM Journal on Scientific Computing*, 28:626–650, 2006.

[47] C. E. Goodyer and M. Berzins. Parallelization and scalability issues of a multilevel elastohydrodynamic lubrication solver. *Concurrency and Computation: Practice and Experience*, 19:369–396, 2007.

[48] M. Griebel, D. Oeltz, and M. Schweitzer. An Algebraic Multigrid Method for Linear Elasticity. *SIAM Journal on Scientific Computing*, 25(2):385–407, 2003.

[49] W. Habchi. *A Full-System Finite Element Approach to Elastohydrodynamic Lubrication Problem: Application to Ultra-Low-Viscosity Fluids*. Phd thesis, University of Lyon, France, 2008.

[50] W. Habchi, D. Eyheramendy, P. Vergne, and G. Morales-Espejel. A Full-System Approach of the Elastohydrodynamic Line/Point Contact Problem. *ASME J. Tribol*, 130, 021501, 2008.

[51] W. Habchi, D. Eyheramendy, P. Vergne, and G. Morales-Espejel. Stabilized Fully-Coupled Finite Elements for elastohydrodynamic Lubrication Problems. *Advances in Engineering Software*, 46(1):4–18, 2012.

[52] B. J. Hamrock and D. Dowson. Isothermal elastohydrodynamic lubrication of point contacts. Part III-fully flooded results. *ASME J. Lubr. Tech.*, 99:264–276, 1977.

[53] B. J. Hamrock and D. Dowson. *Ball Bearing Lubrication*. John Wiley and Sons, New York, 1981.

[54] D. E. Hart. *Adjoint Error Estimation for Elastohydrodynamic Lubrication*. PhD thesis, University of Leeds, Leeds, UK, 2008.

[55] H. Hertz. The contact of elastic solids. *Journal für die reine angew. Math.*, 92:156–171, 1881.

[56] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradient for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[57] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005.

[58] D. C. Hodgson and P. K. Jimack. Efficient Parallel Generation of Partitioned, Unstructured Meshes. *Advances in Engineering Software*, 27:59–70, 1996.

[59] D. L. Hogenboom, W. Webb, and J. A. Dixon. Viscosity of Several Liquid Hydrocarbons as a Function of Temperature, Pressure, and Free Volume. *J. Chem. Phys.*, 46(7):2586–2598, 1967.

[60] M. J. A. Holmes, H. P. Evans, T. G. Hughes, and R. W. Snidle. Transient Elastohydrodynamic Point Contact Analysis Using a New Coupled Differential Deflection Method. Part I: Theory and Validation. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, 217:289–303, 2003.

[61] M. J. A. Holmes, H. P. Evans, T. G. Hughes, and R. W. Snidle. Transient Elastohydrodynamic Point Contact Analysis Using a New Coupled Differential Deflection Method. Part II: Results. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, 217:305–321, 2003.

[62] L. G. Houpert and B.J. Hamrock. Fast Approach for Calculating Film Thicknesses and Pressures in Elastohydrodynamically Lubricated Contacts at High Loads. *ASME Journal of Tribology*, 108:411–420, 1986.

[63] H. S. S. Hsiao, B.J. Hamrock, and J.H. Tripp. Finite Element System Approach to EHL of Elliptical Contacts: Part1- Isothermal Circular Non-Newtonian Formulation. *ASME Journal of Tribology*, 120:695–704, 1998.

[64] T. G. Hughes, C. D. Elcoate, and H. P. Evans. Coupled Solution of the Elastohydrodynamic Line Contact Problem Using a Differential Deflection Method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering*, 214:585–598, 2000.

[65] P. K. Jimack and R. M. Kirby. Towards the development on an h-p-refinement strategy based upon error estimate sensitivity. *Computers and Fluids*, 46:277–281, 2011.

[66] K. L. Johnson. *Contact mechanics*. Cambridge University Press, 1985.

[67] A. C. Jones and P. K. Jimack. An adaptive multigrid tool for elliptic and parabolic systems. *International Journal for Numerical Methods in Fluids*, 47:1123–1128, 2005.

[68] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. Society for Industrial and Applied Mathematics, Philadephia, PA, 2003.

[69] R. Lohner and J. D. Baum. Adaptive h-refinement on 3D unstructured grids for transient problems. *International Journal for Numerical Methods in Fluids*, 14:1407–1419, 1992.

[70] H. Lu. *High Order Finite Element Solution of Elastohydrodynamic Lubrication Problems*. PhD thesis, University of Leeds, Leeds, UK, 2006.

[71] H. Lu, M. Berzins, C. E. Goodyer, and P. K. Jimack. High order discontinuous Galerkin method for elastohydrodynamic lubrication line contact problems. *Communications in Numerical Methods in Engineering*, 21:643–650, 2005.

[72] H. Lu, M. Berzins, C. E. Goodyer, P. K. Jimack, and M. Walkley. Adaptive high-order finite element solution of transient elastohydrodynamic lubrication problems. *Proceedings of the Institution of Mechanical Engineers Part J: Journal of Engineering Tribology*, 220:215–225, 2006.

[73] A. A. Lubrecht. *The Numerical solution of the EHL line and point contact problem using multigrid techniques*. PhD thesis, University of Twente, Endschende, The Netherlands, 1987.

[74] R. Mahmood. *Multilevel Mesh Adaptivity for Elliptic Boundary Value Problems in two and three Space Dimensions*. PhD thesis, University of Leeds, Leeds, UK, 2002.

[75] R. Mahmood and P. K. Jimack. Locally Optimal Unstructured Finite Element Meshes in 3 Dimensions. *Computers and Structures*, 82:2105–2116, 2004.

[76] J. E. Marsden and T. J. R. Hughes. *Mathematical Foundations of Elasticity*. Dover Publications, Inc., New York, 1994.

[77] S. A. Melchior, V. Legat, P. V. Dooren, and A. J. Wathen. Analysis of preconditioned iterative solvers for incompressible flow problems. *International Journal for Numerical Methods in Fluids*, 68(3):269–286, 2012.

[78] M. Moller and D. Kuzmin. Adaptive mesh refinement for high-resolution finite element schemes. *International Journal for Numerical Methods in Fluids*, 52(5):545–569, 2006.

[79] Y. Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis*, 37:123–146, 2010.

[80] E. Nurgat. *Numerical Methods in Lubrication Modelling*. PhD thesis, University of Leeds, Leeds, UK, 1997.

[81] E. Nurgat and M. Berzins. A new relaxation scheme for solving EHL problems. In D. Dowson, C. M. Taylor, T. H. C. Childs, G. Dalmaz, Y. Berthier, L. Flamand, J.-M. Georges, and A. A. Lubrecht, editors, *Proceedings of 23rd Leeds-Lyon Symposium on Tribology*, volume 32 of *Tribology Series*, pages 125–133. Elsevier, 1997.

[82] K. P. Oh and S. M. Rohde. Numerical solution of the point contact problem using the finite element method. *International Journal for Numerical Methods in Engineering*, 11:1507–1518, 1977.

[83] H. Okamura. A Contribution to the Numerical Analysis of Isothermal Elastohydrodynamic Lubrication. *Proc. 9th Leeds-Lyon Symp. Trib.*, 1982(Leeds, UK).

[84] M. E. G. Ong. Uniform refinement of tetrahedron. *SIAM Journal on Scientific Computing*, 15(5):1134–1144, 1994.

[85] A. I. Petrusevich. Fundamental conclusions from the contact-hydrodynamic theory of lubrication. *Izv. Akad. Nauk. SSSR (OTN)*, 2:209, 1951.

[86] M. D. Piggott, P. E. Farrell, C. R. Wilson, G. J. Gorman, and C. C. Pain. Anisotropic mesh adaptivity for multi-scale ocean modelling. *Philosophical Transactions of the Royal Society A*, 367:4591–4611, 2009.

[87] N. Provatas, N. Goldenfeld, and J. Dantzig. Adaptive Mesh Refinement Computation of Solidification Microstructures Using Dynamic Data Structures. *Journal of Computational Physics*, 148:265–290, 1999.

[88] A. P. Ranger and C. M. M. Ettles. The solution of the point contact elastohydrodynamic problem. *Proceedings of the Royal Society of London*, A346:227–244, 1975.

[89] J. K. Reid. On the method of conjugate gradients for the solution of large sparse systems of linear equations. In *Conference on Large Sparse Sets of Linear Equations*, pages 231–254. Academic Press, 1971.

[90] O. Reynolds. On the theory of lubrication and its application to Mr Beauchamp Tower's experiments, including an experimental determination of the viscosity of

olive oil. *Philosophical Transactions of the Royal Society of London*, 177:157–234, 1886.

[91] C. J. A. Roelands. *Correlational Aspects of the Viscosity-Temperature-Pressure Relationship of Lubricating Oils*. PhD thesis, Technische Hogeschool Delft, The Netherlands, 1966.

[92] J. Rosam, P. K. Jimack, and A. Mullis. A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification. *Journal of Computational Physics*, 225(2):1271–1287, 2007.

[93] Y. Saad. *Iterative Methods for Sparse Linear Systems (Second Edition)*. Society for Industrial and Applied Mathematics, Philadephia, 2003.

[94] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[95] J. Schoeberl. Netgen, an advancing front 2d/3d-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997.

[96] W. Speares and M. Berzins. A 3d unstructured mesh adaptation algorithm for time-dependent shock-dominated problems. *International Journal for Numerical Methods in Fluids*, 25(1):81–104, 1997.

[97] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, 2003.

[98] X. Tan, C. E. Goodyer, P. K. Jimack, R. I. Taylor, and M. A. Walkley. Computational Approaches for Modelling Elastohydrodynamic Lubrication Using Multiphysics Software. *Proceedings of the Institution of Mechanical Engineers, Part J, Journal of Engineering Tribology*, 226:463–480, 2012.

[99] C. R. Taylor and J. F. O'Callaghan. A numerical solution of the elastohydrodynamic lubrication problem using finite elements. *Journal of Mechanical Engineering Science*, 14:229–237, 1972.

[100] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.

[101] H. A. van der Vorst. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM J. on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[102] C. H. Venner. *Multilevel Solution of the EHL Line and Point Contact Problems*. PhD thesis, University of Twente, Endschende, The Netherlands, 1991.

[103] C. H. Venner and J. Bos. Effects of lubricant compressibility on the film thickness in EHL line and circular contacts. *Wear*, 173:151–165, 1994.

[104] C. H. Venner and A. A. Lubrecht. *Multilevel Methods in Lubrication*. Elsevier, Netherlands, 2000.

[105] S. R. Wu. A penalty formulation and numerical approximation of the Reynolds-Hertz problem of elastohydrodynamic lubrication. *International Journal of Engineering Science*, 24(6):1001–1013, 1986.

[106] Y. Xiao, S. Shu, and T. Zhao. A geometric-based algebraic multigrid method for higher-order finite element equations in two-dimensional linear elasticity. *Numerical Linear Algebra with Applications,* 16:535–559, 2009.

[107] S. Yasutomi, S. Bair, and W. O. Winer. An Application of a Free Volume Model to Lubricant Rheology I-Dependence of Viscosity on Temperature and Pressure. *ASME J. Tribol.*, 106(2):291–303, 1984.

[108] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, 5th edition, Volume 3: Fluid Dynamics*. Butterworth-Heinmann, Oxford, 2000.

[109] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. Elsevier Butterworth-Heinemann, 2005.

[110] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24(2):337–357, 1987.

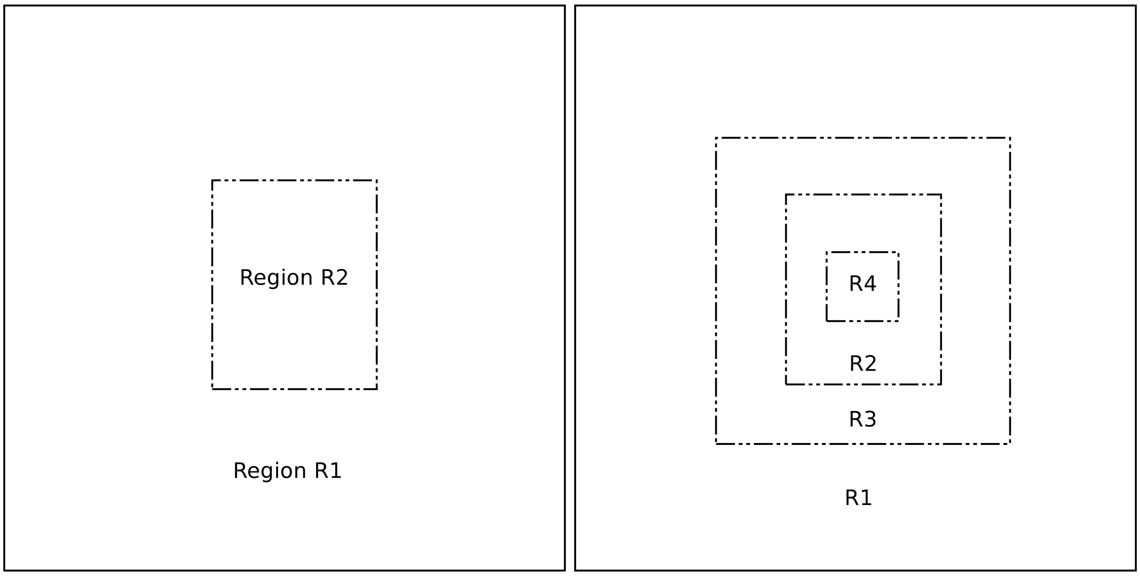
[111] O. C. Zienkiewicz and J. Z. Zhu. The superconvergence patch recovery and a posteriori error estimates. Part 1: The recovery techniques. *International Journal for Numerical Methods in Engineering*, 33:1331–1364, 1992.

[112] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33:1365–1382, 1992.

Appendix A

Elasticity

The shape of an elastohydrodynamic lubrication film depends upon the elastic deformation of the contacting surfaces. In most traditional approaches, e.g. [20, 46, 47, 73, 102, 104], the elastic deformation of contacting surfaces was calculated by using a half space approach, which is based upon an analytical solution of the elasticity equations on a semi-infinite domain. However this approach only provides information about elastic deformation at the surface. On the other hand, the numerical solution of the classical linear elasticity equation on a finite domain provides considerably more information, such as elastic deformation throughout: it can therefore provide engineers with further information such as stresses throughout the contacting elements. However the linear elasticity equation needs to be solved in a 2D domain for line contact problems, and a 3D domain for point contact problems. In a FEM solution of the linear elasticity equations, the more the number of mesh points, the more accurate the solution will be. This however leads to a significant increase in the size of the discrete algebraic system to be solved. It should also be noted that a fine mesh is required in the regions where the solution requires the greatest resolution. Therefore special attention is required to choose different mesh sizes in different parts of the domain, especially in point contact problems in order to get a precise solution with minimal computational cost. The purpose of this experiment is to investigate different mesh sizes in different parts of the 3D domain to get a "sufficiently accurate" elastic deformation with as few finite elements as possible. This investigation is empirical and will be based upon the comparison of different numerical solutions computed on different meshes.



(a) Domain with two sub-regions

(b) Domain with four sub-regions

Figure A.1: A view of the top of the 3D domain

The pressure generated inside the lubricant film is high enough in the contact region to lead to a significant elastic deformation while, on the other hand, the pressure outside the contact region is relatively low and does not lead to a significant elastic deformation. Moreover, the precision of the elastic deformation is important only in the contact region, since this is what affects the Reynolds equation. Hence, we propose that a fine 3D mesh is required in the contact region up to a certain depth, but not in the whole elasticity domain. Nevertheless, a suitable mesh size needs to be defined outside the contact region in order to get a sufficiently accurate solution in the contact region.

In this experiment, the linear elasticity equation is solved on a 3D domain which is a cube of dimension $[-30, 30] \times [-30, 30] \times [-60, 0]$, the motivation behind choosing this dimension comes from the work of Habchi [50, 51], who did experiments on different dimensions and adopt the said dimension in his work.

A view (XY plane, where $Z = 0$) of the top of the domain is shown in Figure A.1. For the sake of simplicity, the notations $R1$ and $R2$ are used to represent different regions in the domain, $R1$ representing the whole domain. $R2$ will represent a central region of dimension $[-4.5, 4.5] \times [-3, 3] \times [-3, 0]$ which contains the contact domain $[-4.5, 1.5] \times [-3, 3]$ (considered in this work) on its top surface. It should be noted that the length of $R2$ is chosen $[-4.5, 4.5]$ instead of $[-4.5, 1.5]$ in order to get a more precise elastic deformation solution. Moreover, in the following section, it will be shown that depth '3' of region $R2$ is good enough to get a sufficiently accurate elastic deformation solution.

Later in this appendix, further regions $R3$ and $R4$ will be defined, whose dimensions will be stated explicitly.

Different resolutions, denoted resolution $1, 2$ and 3, will be used for the mesh sizes in the central region ($R2$ and/or $R4$) throughout this appendix. Resolution 1 is the coarsest resolution and the mesh size for this resolution is about 0.09375 which corresponds to 64×64 points in the contact domain ($[-4.5, 1.5] \times [-3, 3]$), Similarly, resolution 2 and 3 corresponds to 128×128 and 256×256 points in the contact domain respectively. In order to define the most appropriate mesh size away from central region, a number of experiments have been carried out for each region and results are presented in the following sections. For every test case the root mean square error (RMSE) is calculated with respect to the finest case. This will be used to provide an idea about the mesh sizes that need to be adopted for sufficient accuracy in the solution of this problem.

All the meshes used here are generated using NETGEN [95]. In order to get a required local mesh size density, a mesh size file is provided to NETGEN, which specifies 'n' points given by the (x_i, y_i, z_i)-coordinates and parameters h_i, where the mesh size will be reduced at least to h_i. It should be noted that these 'n' points are not necessarily contained in the final mesh.

It should also be noted that an equivalent geometry of a steel to steel contact is considered in this work, and the elastic material contains the total elastic properties of both contact surfaces, and hence the solution will provide the total elastic deformation of both contacting surfaces. The equivalent Young's modulus E_{eq} and Poisson ratio ν_{eq} of equivalent geometry are given by (see Section 2.6)

$$\nu_{eq} = \nu = 0.3$$

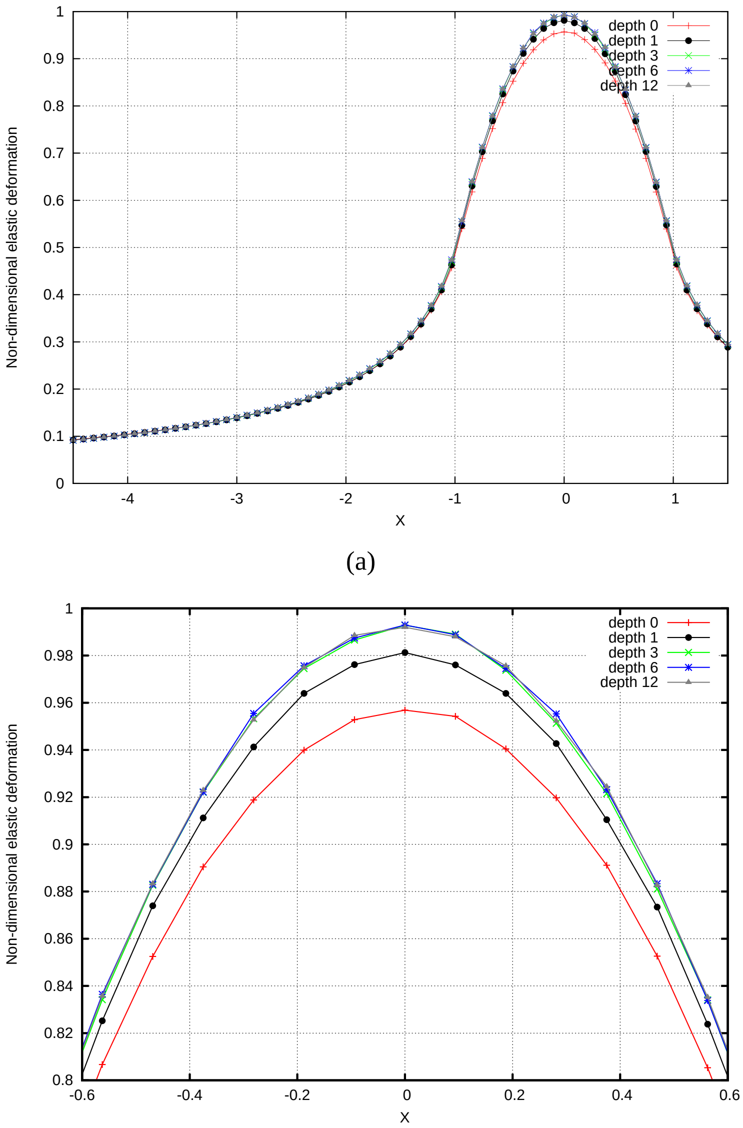
$$E_{eq} = \frac{\pi}{2}(1 - \nu^2)$$

Unless stated otherwise, a Hertzian pressure profile is used to carry out all experiments in this appendix.

A.1 Depth Test for Region $R2$

In this section, the main domain is divided into two regions $R1$ and $R2$ as shown in Figure A.1(a). A fine mesh of resolution 1 has been used in region $R2$ and a relatively coarse mesh is used in region $R1 - R2$. The behaviour of fine mesh in $R2$ down to different depths has been investigated.

It is clear from Figure A.2 that any increase in the depth after '3' does not lead to any



(a)

(b) close-up

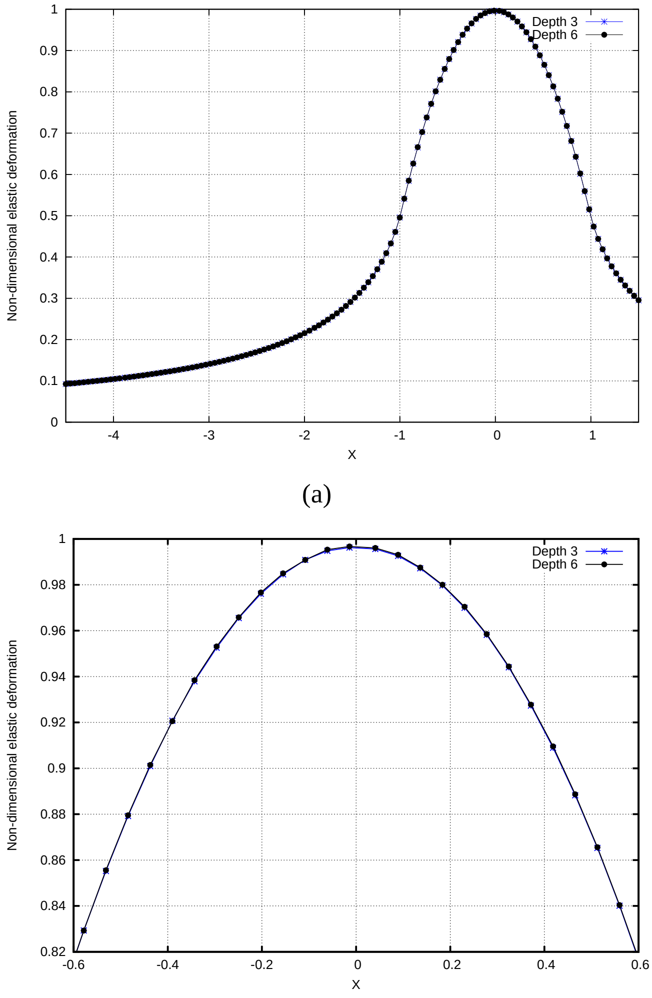
Figure A.2: Effect of using a fine mesh (resolution 1) in $R2$ down to different depths: Accuracy of the elastic deformation solution along the central line (where $Y = 0$).

significant improvement in the elastic deformation result. Moreover, RMSE of each depth case is calculated with respect to a case with depth '12' and are reported in the Table A.1. One can seen that the RMSE of depth 3 is less than 10^{-3}, relative to a non-dimensional deformation of $\mathcal{O}(1)$. Therefore, depth 3 of $R2$ will be adopted in our cases.

A further question arises here is, what is the effect on the depth of region $R2$ with an increase in the fine mesh resolution. For this purpose, another experiment is carried out in which resolution 2 has been used in region $R2$ down to depth 3 and 6. Again Figure A.3

Table A.1: Effect of fine mesh (resolution 1) in $R2$ down to different depths: RMSE of each case is calculated with respect to a case with depth 12.

Case	RMSE
Depth0	8.92×10^{-3}
Depth1	4.55×10^{-3}
Depth3	9.52×10^{-4}
Depth6	3.93×10^{-4}
Depth12	0.0×10^{0}



(a)

(b) close-up

Figure A.3: Effect of using a fine mesh (resolution 2) in $R2$ down to different depths: Accuracy of the elastic deformation solution along the central line (where $Y = 0$).

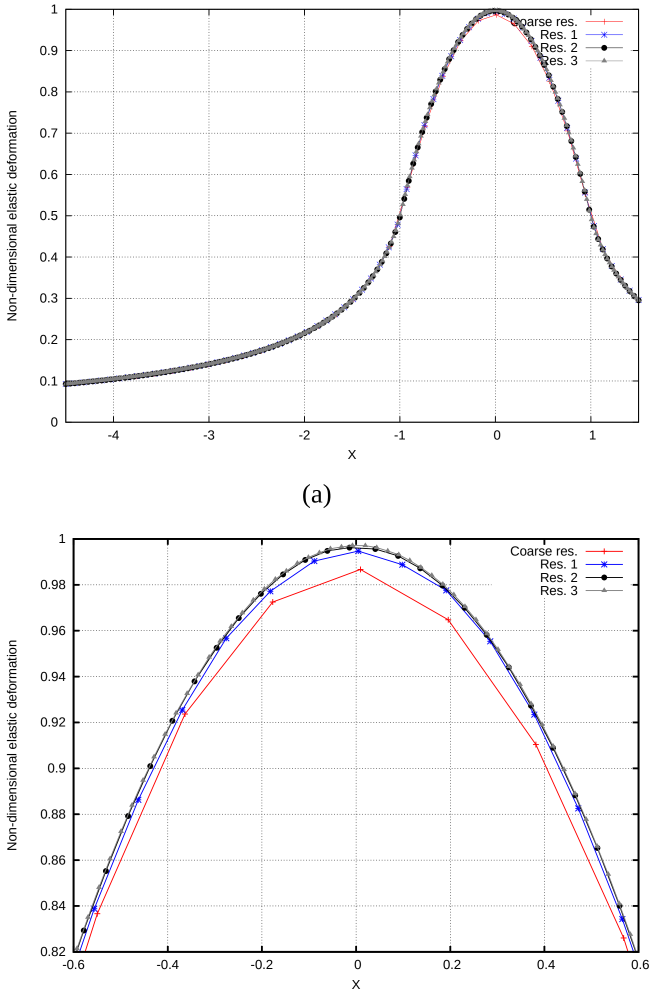
justifies that depth 3 of region $R2$ is working sufficiently well for the higher resolution problem. The RMSE of these two data sets (i.e. comparing the solution with refinement to depth '3' against the solution with refinement to depth '6') is 2.35×10^{-4}, which is again very small. This experiment concludes that a resolution in region $R2$ down to depth '3' is sufficient to get a sufficiently accurate solution. Therefore, depth '3' of region $R2$ will be adopted in these and higher resolution cases.

A.2 Selection of a Reference Solution for Further Analysis

In the previous section, it was suggested that a fine mesh in region $R2$ down to a depth '3' is sufficient to get a sufficiently accurate solution. Before proceeding to the next experiment we need to define a more accurate solution which we named as the "reference solution" in order to get a comparison of different numerical solutions computed on different meshes. This reference solution should be such that any further increase in the resolution of the problem does not lead to a significant improvement in the solution. For this purpose, a test has been undertaken to check the convergence of solutions with increases in the resolution. Again, the main domain is divided into two regions $R1$ and $R2$ as shown in Figure A.1(a). The coarsest resolution we considered in this analysis uses a mesh size about '0.1875' everywhere in region $R2$ and a mesh size '0.5' outside of region $R2$. In the other three cases, the resolution is increased everywhere in region $R2$ and uses the mesh sizes which correspond to resolution 1, 2 and 3. These four cases are named as 'coarse resolution', 'resolution 1', 'resolution 2' and 'resolution 3' respectively. It should be noted that resolution 3 case leads to $15\,802\,299$ nodes in the mesh (over 45 million unknowns), which makes the problem very large and around '110GB' memory was required to solve this problem. These four results are plotted together in Figure A.4, while the RMSE of each resolution with respect to finest resolution is provided in Table A.2. Figure A.4 shows that results on resolution 2 and 3 are very close and suggests that a further increase in the resolution will not significantly improve the solution. This fact can also be justified from Table A.2, which reveals that RMSE is well under 10^{-3}, even for resolution 1. Hence, we conclude that resolution 1 is sufficiently accurate in this refined region, whilst the resolution 3 case provides a particularly accurate solution, which can be used as a reference solution to do further analysis tests.

A.3 Mesh Sizes Tests (Meshing Strategy 1)

In the previous section, we have defined a reference solution for further comparisons. In the convergence tests (see previous section), we increased the resolution everywhere in



(a)

(b) close-up

Figure A.4: Effect of using different resolutions in region $R2$: Accuracy of the elastic deformation solution shown along the centreline (where $Y = 0$).

Table A.2: RMSE of different resolution cases in region $R2$ calculated with respect to the finest resolution case.

Case	RMSE
Coarse Res.	3.06×10^{-3}
Res. 1	8.51×10^{-4}
Res. 2	2.10×10^{-4}
Res. 3	0.0×10^{0}

region $R2$ and moreover, we have used a relatively small mesh size (0.5) in the region $R1 - R2$. However, all four extreme cases in the previous section lead to large numbers of nodes, which are respectively, $3\,302\,424$, $3\,505\,403$, $4\,950\,405$ and $15\,802\,299$. This results in a very large problem to solve (even at resolution 1), which is very expensive from a computational point of view. For a sufficiently accurate solution, the small mesh size everywhere in the domain is not necessary, while, keeping in mind, the region of interest. Hence a number of questions arise as to how to improve the efficiency:

1. Is the mesh size '0.5' enough in the region $R1 - R2$ or can a coarser mesh size be used without significantly affecting the accuracy of solution?

2. Does the mesh size used in $R1 - R2$ need to be the same everywhere in the region $R1 - R2$ or can this be adapted only in a certain region around region $R2$?

3. Does the mesh size at each resolution need to be adopted everywhere in region $R2$, if no, where should this resolution be adopted inside $R2$ and what should be the dimension of that region and finally, what mesh size should be used in the remaining region of $R2$?

These questions lead to the definition of two more sub-regions, namely $R3$ and $R4$, outside and inside of region $R2$, respectively. This decomposition of the whole domain can be viewed in Figure A.1(b). To answer all these questions, a number of further tests have been carried out and results are discussed in the following sub-sections.

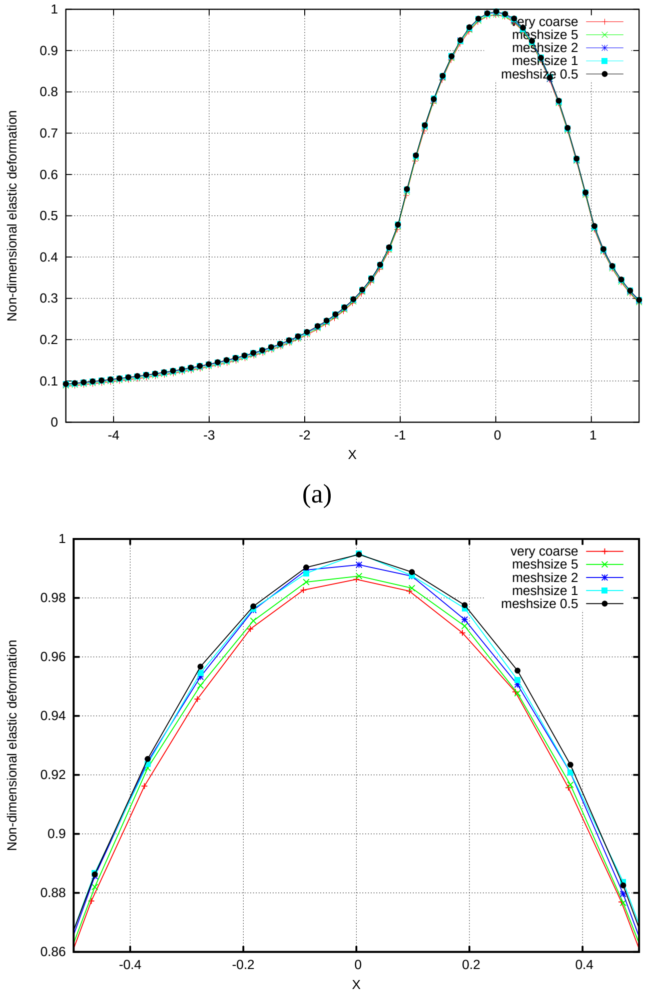
A.3.1 Resolution 1

In this section, all tests use resolution 1 in the most refined region, and have been carried out in order to define the most appropriate mesh sizes away from the most central region.

A.3.1.1 Effect of Mesh Size Outside of Region $R2$

In this experiment, the main domain is divided into two regions $R1$ and $R2$. The mesh size in $R2$ with depth 3 is kept constant (at resolution 1), while in $R = (R1 - R2)$, different mesh sizes have been tested. It is clear from the Figure A.5, that any decrease in the mesh size below 1.0 does not lead to any significant improvement in the solution. Moreover, Table A.3 reveals that RMSE of mesh size 1.0 and 0.5 is almost the same (less than 10^{-3} error relative to an $\mathcal{O}(1)$ displacement). Therefore, the mesh size 1.0 in region $R1 - R2$ will be adopted in our work for the current resolution.

For a $60 \times 60 \times 60$ domain, the mesh size 1.0 leads to roughly 216000 nodes in $R1 - R2$, which is still quite large in number. So instead of adopting this mesh size in the outer region $R1 - R2$, we carried out a further test, where we have introduced a region $R3$ outside of $R2$. A mesh size of 1.0 is adopted in $R3 - R2$, while a size of 5.0 is used in the remaining region outside of $R3$. This test is carried out on different dimensions of $R3$.



(a)

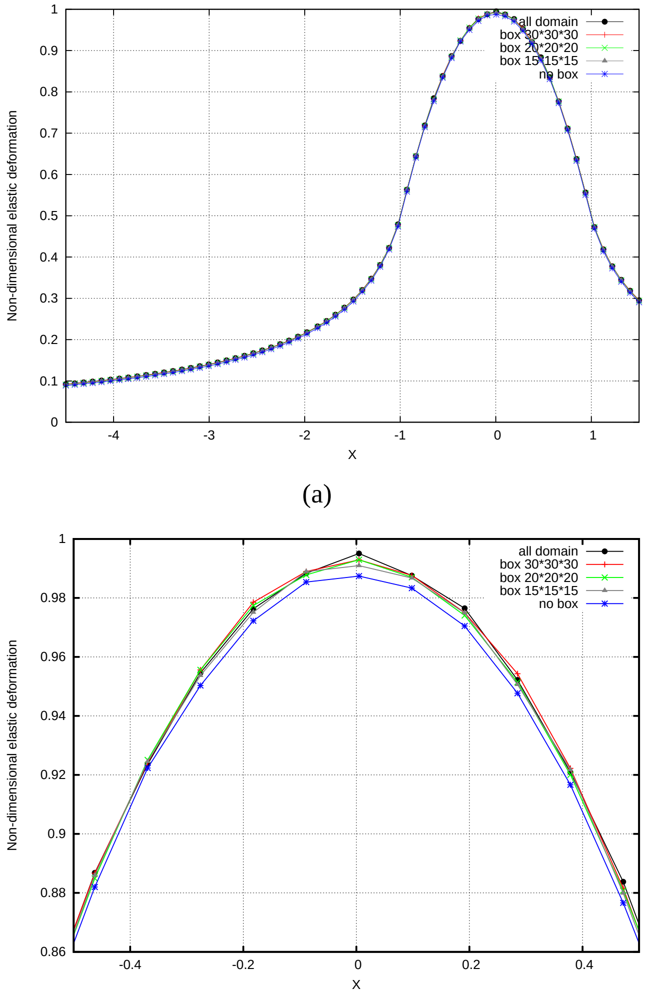
(b) close-up

Figure A.5: Effect of mesh size outside of region $R2$: Accuracy of the elastic deformation profile (shown along the centreline).

Table A.3: Effect of mesh size outside of region $R2$: RMSE of each case is calculated with respect to the reference solution.

Case	RMSE
Very coarse	6.67×10^{-3}
Mesh-size 5.0	4.62×10^{-3}
Mesh-size 2.0	1.81×10^{-3}
Mesh-size 1.0	8.47×10^{-4}
Mesh-size 0.5	8.51×10^{-4}

There are two extreme cases, one with “no box” meaning a mesh size of 5.0 is adopted in the remaining domain outside of $R2$ and the other with $R3$ overlapping $R1$, i.e. a mesh size of ‘1.0’ is used in the entire region outside of $R2$. These results are plotted together



(a)

(b) close-up

Figure A.6: Effect of a fixed mesh size in the region $R3 - R2$ with different dimensions of $R3$: Accuracy of the elastic deformation profile (shown along the centreline).

Table A.4: Effect of a fixed mesh size in the region $R3 - R2$ with different dimensions of $R3$: RMSE of each case is calculated with respect to the reference solution.

Case	RMSE
No box	4.62×10^{-3}
Box $15 \times 15 \times 15$	1.88×10^{-3}
Box $20 \times 20 \times 20$	1.45×10^{-3}
Box $30 \times 30 \times 30$	9.42×10^{-4}
All domain	8.47×10^{-4}

in Figure A.6, which shows that choices $20 \times 20 \times 20$ and $30 \times 30 \times 30$ works fine, but on the other hand a careful look on the Table A.4 suggests that $30 \times 30 \times 30$ is a better choice than $20 \times 20 \times 20$ for the current resolution case (the former giving an error of less

Table A.5: Effect of using mesh sizes $1.0h, 1.5h$ and $2.0h$ in $R2 - R4$ (where h is mesh size corresponding to resolution 1): RMSE of each case is calculated with respect to the reference solution.

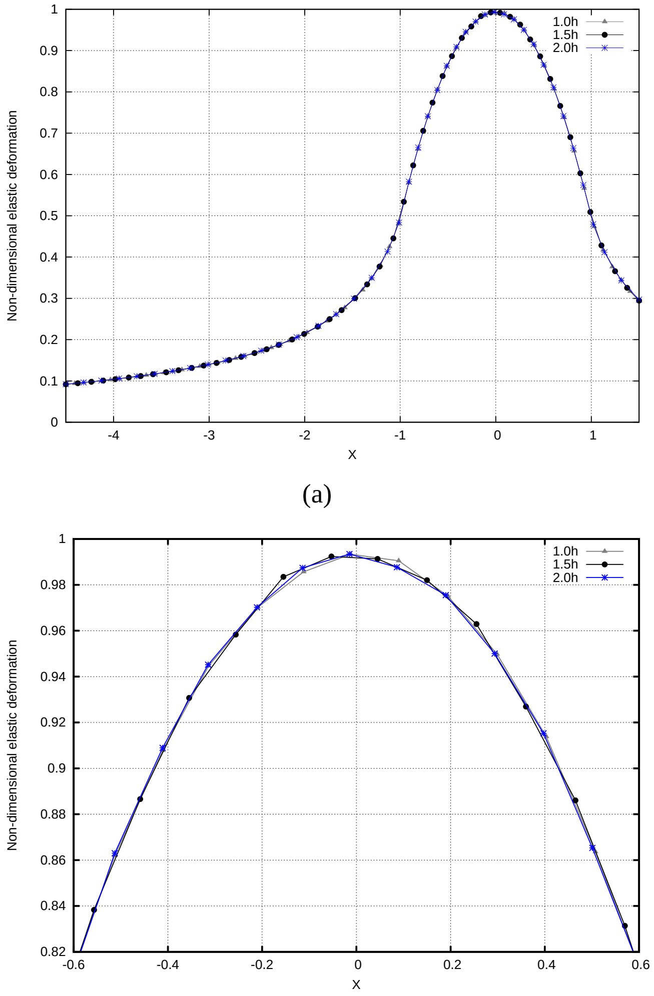
Case	RMSE
1.0h in $R2 - R4$	1.48×10^{-3}
1.5h in $R2 - R4$	1.71×10^{-3}
2.0h in $R2 - R4$	1.78×10^{-3}

than 10^{-3}, while the latter gives an error which is slightly larger).

A.3.1.2 Coarsening of Mesh in the Central Region

So far the mesh size in a central region $R2$ was kept constant and different mesh sizes were tested outside this region, in the remaining domain. In all experiments, the mesh size in the central region $R2$ was chosen 0.09375. This mesh size, although not very small, still leads to a relatively large number of points (roughly 147456) in the central region $R2$. In this test, a region $R4$ of dimension $[-1, 1] \times [-1, 1] \times [-2, 0]$ is defined inside the central region $R2$ and the mesh size 0.09375 is adopted in this new region $R4$. Now in the region $R2 - R4$, $1, 1.5$ and 2 times of this mesh size is adopted. The number of nodes contained in the mesh for these three cases are respectively, 303591, 255578 and 116548.

The RMSE of all these three cases (with respect to the finest reference case) are given in Table A.5. One can observe that, switching from $1.0h$ to $1.5h$ in $R2 - R4$, although not significantly affecting the number of nodes, does increase the RMSE by a modest amount. On the other hand, switching from $1.5h$ to $2.0h$ in $R2 - R4$ does significantly affect the number of nodes, but with only a very small further increase in the RMSE. Moreover, the first case is the same as case-4 of Table A.4, the only difference is the detail of the mesh generation, i.e. in the former case mesh sizes are provided separately for region $R2 - R4$ and $R4$, while in the later case (case-4 of Table A.4), mesh size was specified for whole region $R2$. This led to generation of two different meshes leaving a small effect on difference in nodes which is 177, but the RMSE is changed from 9.42×10^{-4} to 1.48×10^{-3}. So keeping in mind this variation in the solution, the choice of $2.0h$ in $R2 - R4$ seems to be a better choice, as this is leading to a large decrease in number of nodes about 187043 without significantly affecting the RMSE. Figure A.7 shows a graphical comparison of all three cases. It should be noted that, adopting different mesh sizes in $R2 - R4$ is not significantly affecting the overall solution, and a small variation in the individual nodal values is observed around the centre.



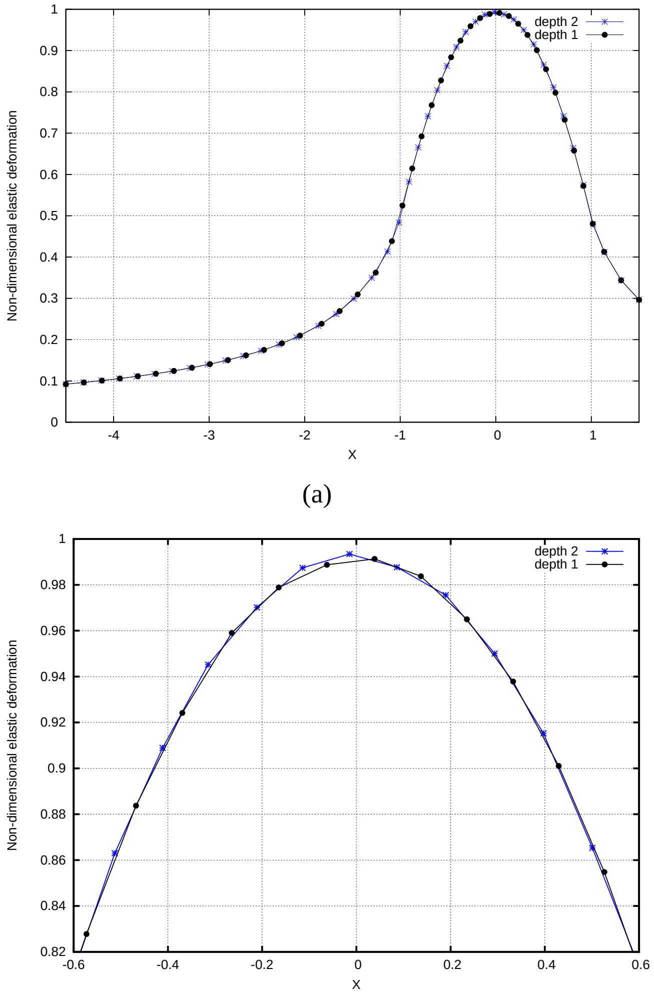
(a)

(b) close-up

Figure A.7: Effect of using mesh sizes $1.0h, 1.5h$ and $2.0h$ in $R2 - R4$ (where h is mesh size corresponding to resolution 1): Accuracy of the elastic deformation profile (shown along the centreline).

A.3.1.3 A Note on the Reduction in the Depth of $R4$

In this experiment, the effect of halving the depth of $R4$ is discussed, i.e. mesh size 0.09375 is adopted in $R4$ down to depth 1 instead of 2. This lead to a small reduction in number of nodes from 116548 to 110984. Table A.6 shows that, it has slightly affected the RMSE. Moreover, Figure A.8 shows that a small variation is seen near the centre. Being a coarse resolution case, decrease in depth of region $R4$ is not significantly affecting the number of nodes contained in resultant mesh. Therefore, for the current case, there is no significant gain in reducing the depth of $R4$.



(a)

(b) close-up

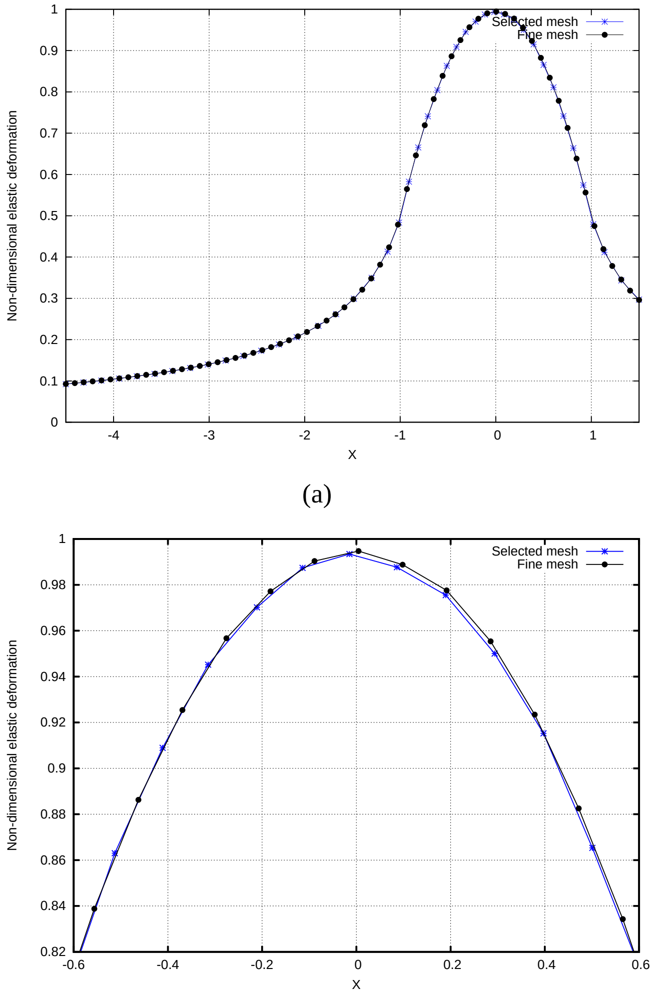
Figure A.8: Effect of using a mesh size corresponding to resolution 1 in $R4$ down to different depths: Accuracy of the elastic deformation profile (shown along the centreline).

Table A.6: Effect of using a mesh size corresponding to resolution 1 in $R4$ down to different depths: RMSE of each case is calculated with respect to the reference solution.

Case	RMSE
Depth 2	1.78×10^{-3}
Depth 1	2.14×10^{-3}

A.3.1.4 Conclusion and Overall Comparison

So far we have divided the whole domain into four regions, the first is the most central region, $R4$, and has dimension $[-1,1] \times [-1,1] \times [-2,0]$, where a mesh size 0.09375 is defined. Second region $R2$ is of dimension $[-4.5,4.5] \times [-3,3] \times [-3,0]$, and a mesh size 0.1875 is selected in $R2 - R4$. In the third region $R3$, whose dimension is $[-15,15] \times$



(a)

(b) close-up

Figure A.9: A comparison of elastic deformation profiles using fine and selected mesh of resolution 1.

$[-15, 15] \times [-30, 0]$, a mesh size 1.0 is used in $R3 - R2$. And the fourth region is the remaining region of the domain, where a mesh size 5.0 is used. The selection of this mesh leads to '116548' nodes in the domain. On the other hand, the finest mesh, with which, we are comparing the solution is split into two main regions: the first region is of dimension $[-4.5, 4.5] \times [-3, 3] \times [-3, 0]$, where the mesh size 0.09375 has been used; and second region is the remaining region of the domain, where a mesh size of 0.5 is used. This finest mesh leads to 3505403 nodes in the domain. A comparison of results on both meshes is given in Figure A.9, and it can be seen that both results are very close, with only a relatively small loss of precision with a huge reduction in the number of points, when switching from a very fine mesh to the selected mesh.

Table A.7: Set of mesh cases for resolution 2: details of mesh sizes used in different regions of the domain.

Case	$R4$	$R2 - R4$	$R3 - R2$	$R1 - R3$
Case-1	$\mathbf{0.046875}$	0.1875	1.0	5.0
Case-2	0.046875	$\mathbf{0.09375}$	1.0	5.0
Case-3	0.046875	0.09375	$\mathbf{0.5}$	5.0
Case-4	0.046875	0.09375	0.5	$\mathbf{2.0}$
Case-5	0.046875	0.09375	0.5	$\mathbf{1.0}$
Fine-case	0.046875	$\mathbf{0.046875}$	0.5	$\mathbf{0.5}$

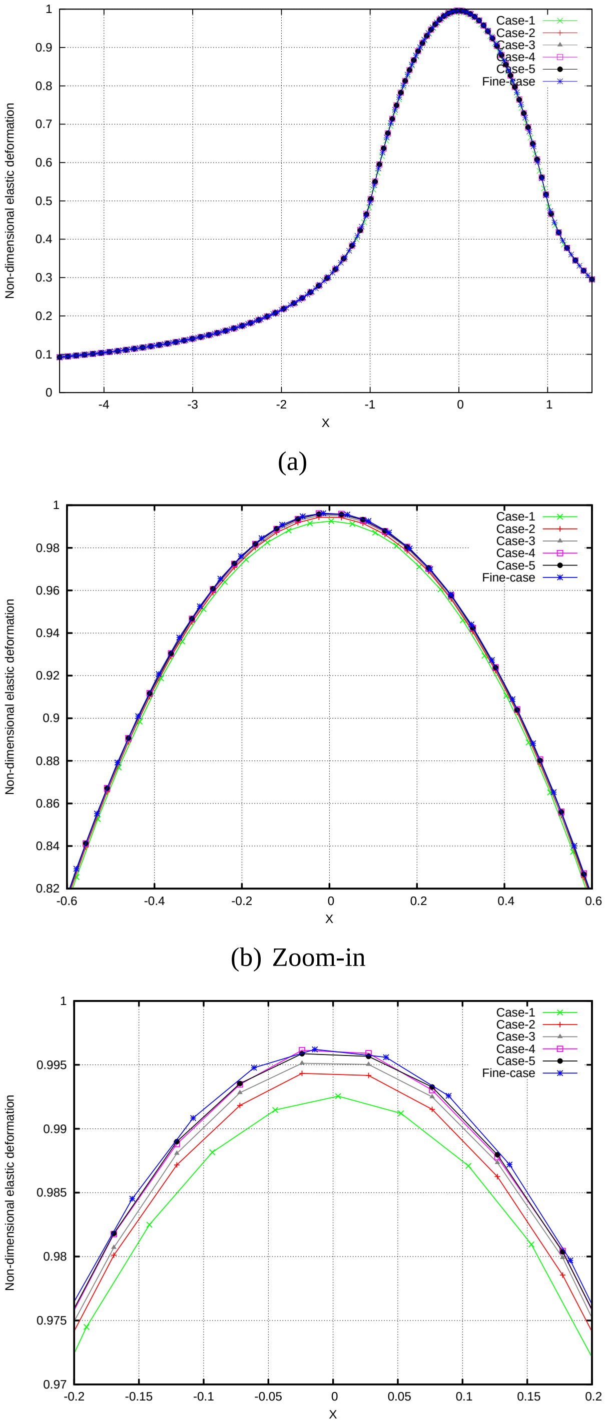
A.3.2 Effect on Mesh Sizes with Increase in Resolution in the Central Region

A.3.2.1 Resolution 2

Recall from previous section that, we divided the whole domain into four regions, where we have used mesh sizes, 0.09375, 0.1875, 1 and 5, from central to outer regions, respectively. The mesh sizes defined (in different regions) were corresponding to resolution 1, which we have adopted in the central most region. In this section, we discuss the effect of increasing the resolution in the central region upon the mesh sizes defined before. In order to proceed, first of all, we divide the whole domain into the following four regions:
$R1$ being the whole domain
$R3$ is a region of dimensions $[-10, 10] \times [-10, 10] \times [-20, 0]$
$R2$ is a region of dimensions $[-4.5, 4.5] \times [-3, 3] \times [-3, 0]$
$R4$ is a region of dimensions $[-1, 1] \times [-1, 1] \times [-1, 0]$
It should be noted that, dimensions of $R3$ and $R4$ are different from those used in the previous section, and these are selected as a result of different experiments done likewise in the previous section.

The first case considered here uses the mesh sizes selected in the previous section (from central to outer region) except the resolution 1 is replaced with resolution 2 in the central region $R4$. In the other cases, the mesh sizes are decreased in the outer regions to check their effect on the accuracy of solution. The mesh sizes, we used for each case (considered here) are given in the Table A.7. While defining the next case, changes made in the previous case are in bold face.

Results corresponding to these cases are plotted together in Figure A.10 and RMSE for all these cases with respect to reference solution case (see Section A.2) are given in Table A.8. Figure A.10 reveals that results corresponding to case-4, case-5 and fine-case are very close to each other, i.e. switching to case-5 and the fine-case does not make a noticeable difference in the solution. Moreover, from Table A.8, it is clear that RMSE



(a)

(b) Zoom-in

(c) Further zoom-in

Figure A.10: A comparison of elastic deformation profiles (for resolution 2) corresponding to different mesh cases defined in Table A.7

Table A.8: RMSE of solutions (computed on different mesh cases defined in Table A.7) with respect to reference solution.

Case	nodes	RMSE
Case-1	113412	3.60×10^{-3}
Case-2	305095	1.78×10^{-3}
Case-3	421270	1.10×10^{-3}
Case-4	462344	5.23×10^{-4}
Case-5	794376	5.23×10^{-4}
Fine-case	4950405	2.10×10^{-4}

Table A.9: Set of mesh cases for resolution 3: details of mesh sizes used in different regions of the domain.

Case	$R4$	$R2 - R4$	$R3 - R2$	$R1 - R3$	Remarks
Case-1	$\mathbf{0.02344}$	0.09375	0.5	2.0	-
Case-2	0.02344	$\mathbf{0.04688}$	0.5	2.0	-
Case-3	0.02344	0.04688	0.5	$\mathbf{1.0}$	-
Case-4	0.02344	0.04688	0.5	$\mathbf{0.5}$	$R4$ with depth ‘2’
Fine-case	0.02344	$\mathbf{0.02344}$	0.5	0.5	-

of case-4 is not affected, while switching to case-5. Although, switching to the fine-case, with $4\,950\,405$ nodes in the mesh, does decrease the RMSE however it is already at an acceptable value before this huge increase in the size of problem. Therefore, it is suggested that mesh sizes defined in case-4 give a sufficiently accurate solution (RMSE well under 10^{-3} relative to an $\mathcal{O}(1)$ displacement) with less computational cost, as this case leads to only $462\,344$ nodes in the mesh.

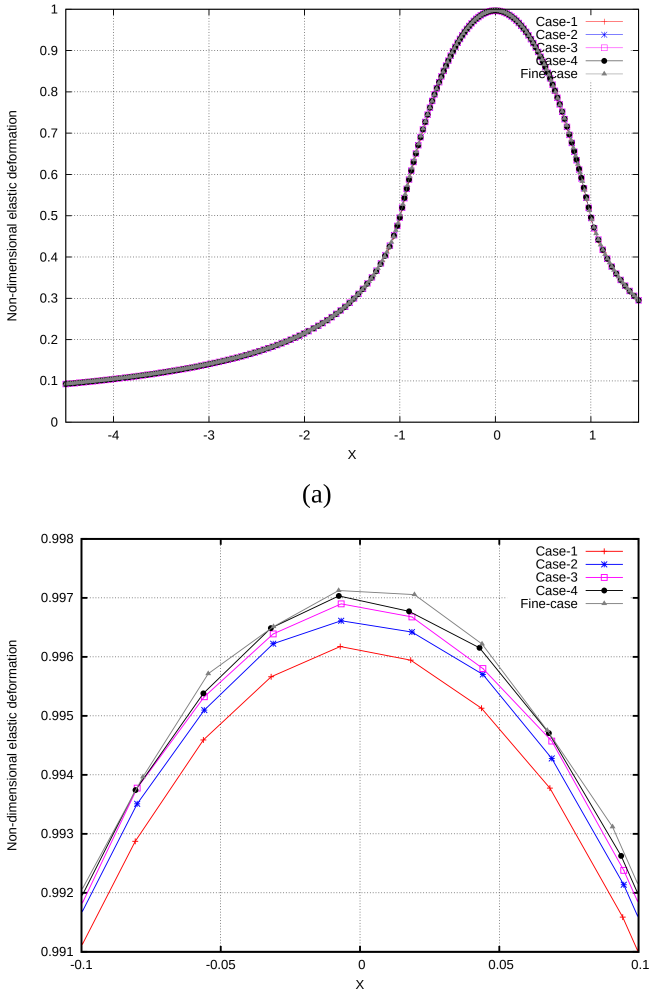
A.3.2.2 Resolution 3

In this section, we discuss the effect of further increase in resolution (in the central region) upon the mesh sizes defined in the previous section A.3.2.1. This experiment uses the same regions defined before. Again the first case considered here uses the mesh sizes as the preferred case (case 4) selected in the previous section A.3.2.1 (from central to outer region), except the resolution 2 is replaced with resolution 3 in the central region $R4$. In other cases, the mesh sizes are decreased in the outer regions to check their effect on the accuracy of solution. The mesh sizes used for each case (considered here) are given in the Table A.9. While defining the next case, changes made in the previous case are in bold face.

Results corresponding to these cases are plotted together in Figure A.11 and the RMSE for all these cases (with respect to reference case, see Section A.2) along with number of nodes in each mesh are given in Table A.10. Figure A.11 reveals that results corresponding to case-3, case-4 and the fine case are very close to each other, i.e. switch-

Table A.10: RMSE of solutions (computed on different mesh cases defined in Table A.9) with respect to reference solution.

Case	nodes	RMSE
Case-1	782270	8.28×10^{-4}
Case-2	2172986	3.72×10^{-4}
Case-3	2510252	1.82×10^{-4}
Case-4	5664718	1.58×10^{-4}
Fine-case	15802299	0.00×10^{0}



(a)

(b) Zoom-in

Figure A.11: A comparison of elastic deformation profiles for resolution 3 corresponding to different mesh cases defined in Table A.9

ing to case-4 or the fine-case from case 3 does not make a noticeable difference in the solution. Moreover, from Table A.10, it is clear that RMSE of case-3 is only slightly affected when switching to case-4 with $5\,664\,718$ (more than double) nodes in the mesh.

Table A.11: Set of mesh cases for resolution 3: details of mesh sizes used in different regions of the domain.

Case	Nodes	Different regions with mesh sizes					
		$R4$	$R5-R4$	$R2-R5$	$R3-R2$	$R6-R3$	$R1-R6$
Case-1	2510252	0.02344	0.04688		0.5	1.0	
Case-2	2093579	0.02344	0.04688	**0.0625**	0.5	1.0	**2.0**
Case-3	1192335	0.02344	0.04688	**0.09375**	0.5	1.0	2.0

Table A.12: RMSE of solutions (using different mesh cases defined in Table A.11) with respect to reference solution.

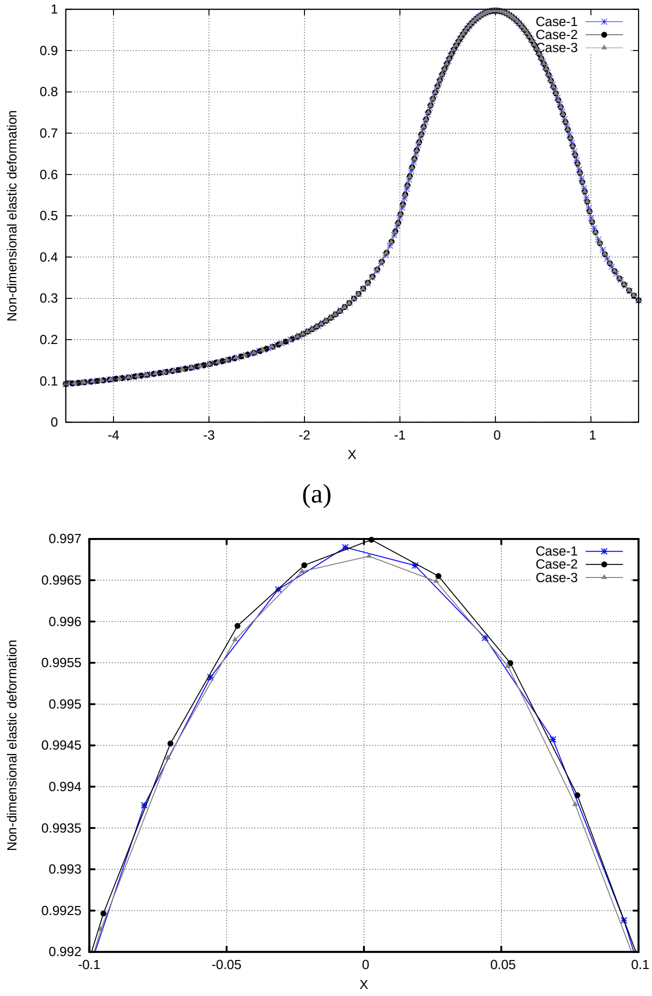
Case	RMSE
Case-1	1.82×10^{-4}
Case-2	1.92×10^{-4}
Case-3	2.74×10^{-4}

In other words, there is no significant gain in accuracy with a huge increase in the size of problem. Therefore, it is suggested that mesh sizes defined in case-3 give a sufficiently accurate solution (RMSE well close to 10^{-4} relative to an $\mathcal{O}(1)$ displacement) with less computational cost than case-4 and fine-case.

As we have seen case-3 is giving a considerably accurate solution, however it is leading to a number of nodes which is becoming relatively large. This is due to the decrease in the mesh sizes in regions $R4$, $R2-R4$ and $R1-R3$. In order to attempt to make this problem smaller, we carried out an experiment where we have defined two more sub-regions namely $R5$ and $R6$. In other words, we divide the whole domain into the following six regions:

$R1$ being the whole domain
$R6$ is a region of dimensions $[-20, 20] \times [-20, 20] \times [-40, 0]$
$R3$ is a region of dimensions $[-10, 10] \times [-10, 10] \times [-20, 0]$
$R2$ is a region of dimensions $[-4.5, 4.5] \times [-3, 3] \times [-3, 0]$
$R5$ is a region of dimensions $[-2, 2] \times [-2, 2] \times [-2, 0]$
$R4$ is a region of dimensions $[-1, 1] \times [-1, 1] \times [-1, 0]$

It should be noted that the region $R2 - R4$ is split into further sub-regions $R2 - R5$ and $R5 - R4$, similarly $R1 - R3$ is split into further sub-regions $R1 - R6$ and $R6 - R3$. Now, instead of adopting the mesh size 0.04688 everywhere in $R2 - R4$, we will only adopt this mesh size in the sub-region $R5 - R4$ and use a slightly larger mesh size in the second sub-region $R2 - R5$. Similarly, the mesh size 1.0 will be adopted in $R6 - R3$ and a larger mesh size in $R1 - R6$. A set of cases considered in this experiment are given in the Table A.11 where case-1 is the preferred case selected in the previous experiment, i.e.



(a)

(b) Zoom-in

Figure A.12: A comparison of elastic deformation profiles for resolution 3 corresponding to different mesh cases defined in Table A.11

case-3 in Table A.9.

All of these cases are plotted together in Figure A.12 and the RMSE of these cases are given Table A.12. Figure A.12 shows that all results are close to each other i.e. switching to case-3 did not lead to a significant loss in the accuracy of the solution. Moreover, Table A.12 shows that the RMSE is only slightly affected when switching to case-3. However, it should be noted that case-3 leads to only 1192335 nodes in the mesh which is less than 50% of the nodes contained in the mesh of case-1. Therefore, from a computational point of view, case-3 is a suitable choice of mesh that leads to a relatively small problem and a sufficiently accurate solution.

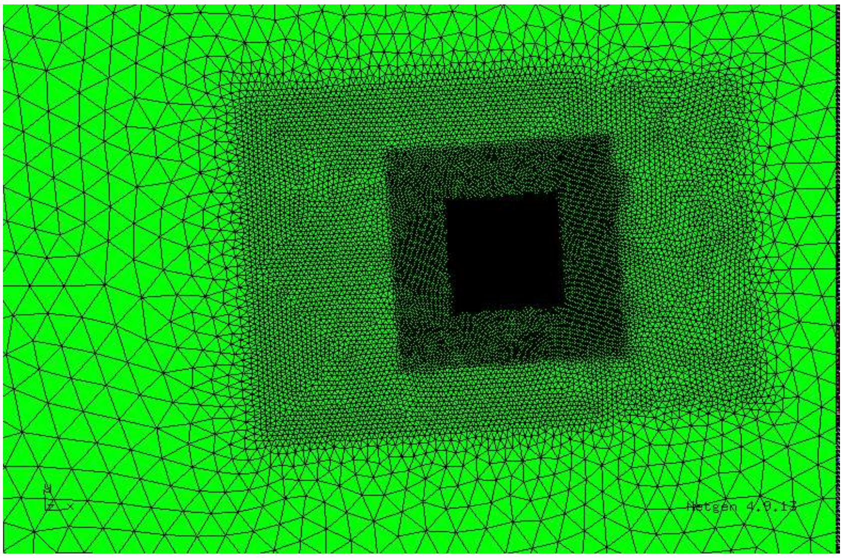


Figure A.13: A top view of selected mesh case of resolution 3 (using meshing strategy 1).

A.3.3 A Note on Generated Meshes

Throughout this section, a number of experiments have been performed to define appropriate mesh sizes away from the central region when using different resolutions in the central region. This led to the definition of different sub-regions in the domain and it is observed that increasing the resolution in the central region has also affected the mesh sizes necessary in outer regions, especially those closest to the central region, i.e. this effect is smaller on the mesh sizes which are away from central region. A top view of the mesh generated for resolution 3 is given in Figure A.13. It can clearly be seen that "NETGEN" adopted the specified mesh size everywhere in a particular region.

One observation to come from the last experiment is that adopting a particular mesh size everywhere in a particular region is not necessary, and it leads to too many nodes in the mesh. In other words splitting a region into further sub-regions and varying mesh sizes within these regions can lead to a large decrease in number of nodes without significantly affecting the accuracy of solution. However, defining more and more sub-regions along with appropriate mesh sizes is not a practical task. In the next section, we will therefore adopt a different strategy to generate such meshes (with "NETGEN") which will cover this aspect and there will be no need to define further sub-regions.

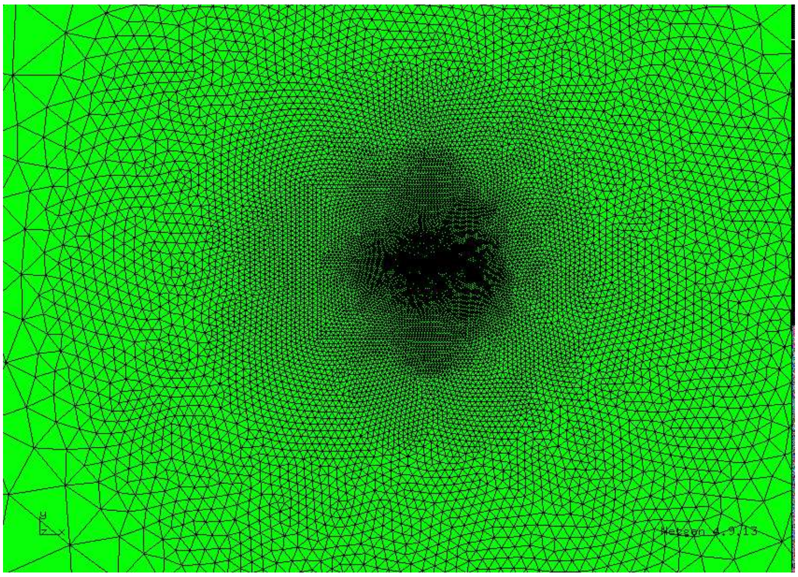
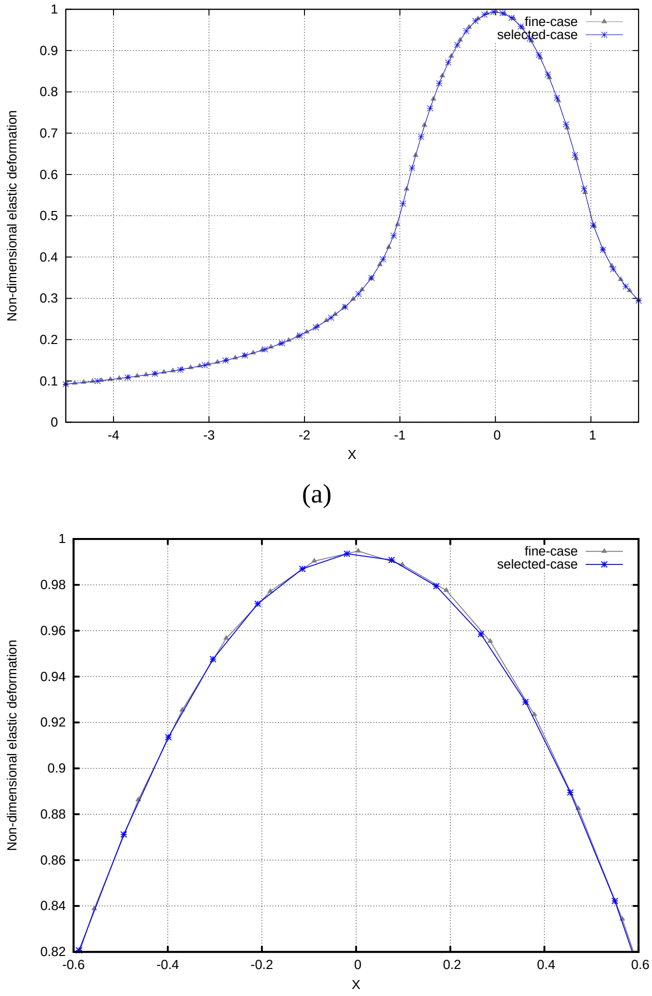


Figure A.14: A top view of a typical mesh generated by using meshing strategy 2.

A.4 Alternative Meshes (Meshing Strategy 2)

In this section, the main purpose is to generate meshes which automatically get coarser and coarser as we move away from the central region and there should be no need to define further sub regions. In the following we generated such a mesh, which uses only two semi-spherical regions of radii ‘1’ and ‘15’ respectively. The choice of these radii is based upon the experiments done in the previous sections. A selected resolution is specified for all points inside the semi-spherical region of radius ‘1’ (region-1) and for points on the curved boundary of semi-spherical region of radius ‘15’ (region-2), a suitable mesh size is defined. For all other interior points of region-2, a linear interpolation is used to define a mesh size for each point. Providing this mesh size file to “NETGEN” leads to the mesh we are acquiring. A top view of such typical mesh generated is given in the Figure A.14. Readers are reminded again that these selections are based upon our previous experimentation. It should be noted that in the remaining region of domain (outside of region-2), different large mesh sizes are used depending upon the resolution used in the central most region. In the following subsections, we obtain a resulting mesh for each resolution and this is compared with its selected case which has been obtained in the previous section and the RMSE of each case is calculated with respect to the reference solution (see Section A.2).



(a)

(b) Zoom-in

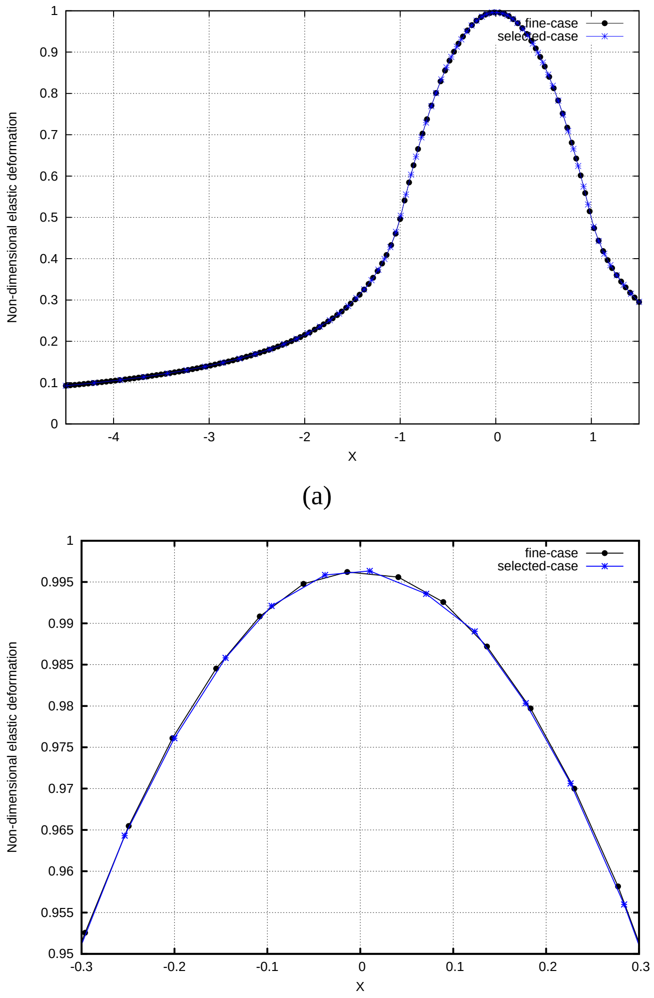
Figure A.15: A comparison of elastic deformation profiles computed on the selected mesh (meshing strategy 2) and fine mesh of resolution 1.

Table A.13: RMSE of solutions, on current selected mesh (meshing strategy 2) and the previous selected mesh (meshing strategy 1) of resolution 1, with respect to the reference solution.

Case	nodes	RMSE
Current-case	102673	1.91×10^{-3}
Previous-case	116548	1.78×10^{-3}

A.4.1 Resolution 1

In this test, resolution 1 is adopted in region-1 and a mesh size ‘1.0’ is specified for the points on the curved boundary of region-2. For all interior points of region-2, a linear interpolation is used to define their corresponding mesh sizes. And finally, in the remain-



(a)

(b) Zoom-in

Figure A.16: A comparison of elastic deformation profiles computed on the selected (meshing strategy 2) and fine mesh of resolution 2.

ing domain, a mesh size ‘4.0’ is used everywhere. The result obtained is plotted together with the fine-case of current resolution (see previous section) in the Figure A.15, while the corresponding RMSE along with the RMSE of previous selected case is given in the Table A.13. Figure A.15 reveals that the result obtained is very close to the one on fine mesh. Moreover, Table A.13 shows that the current case leads to a smaller number of nodes, however, the RMSE of the current case is only slightly larger than that we had in the previous selected case.

Table A.14: RMSE of solutions, on current selected mesh (meshing strategy 2) and the previous selected mesh (meshing strategy 1) of resolution 2, with respect to the reference solution.

Case	nodes	RMSE
Current-case	209665	5.37×10^{-4}
Previous-case	462344	5.23×10^{-4}

Table A.15: RMSE of solutions, on current selected mesh (meshing strategy 2) and the previous selected mesh (meshing strategy 1) of resolution 3, with respect to the reference solution.

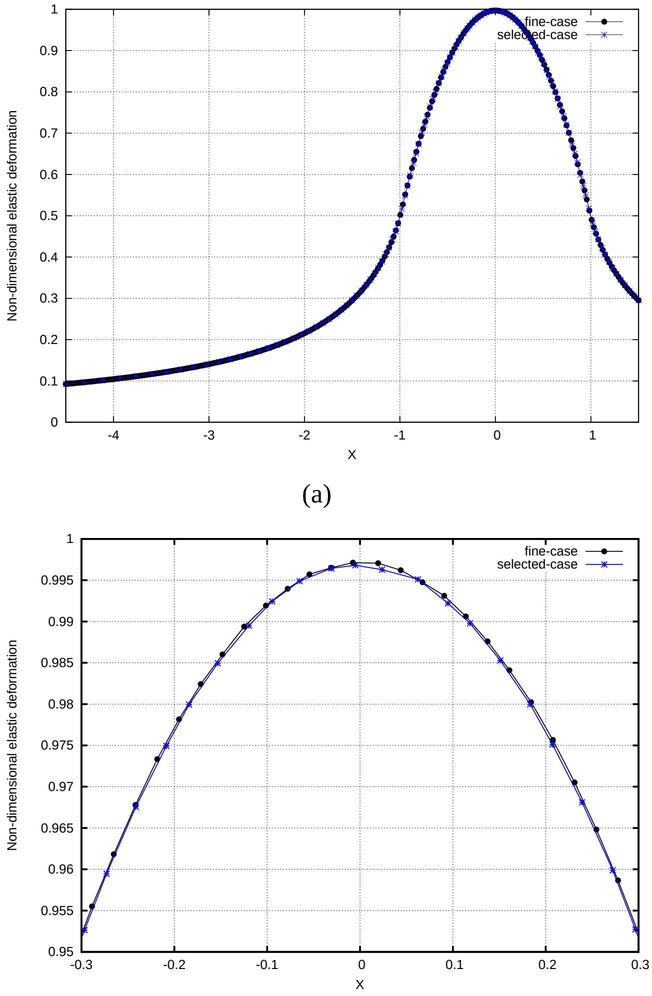
Case	nodes	RMSE
Current-case	638298	2.80×10^{-4}
Previous-case	1192335	2.74×10^{-4}

A.4.2 Resolution 2

In this test, resolution 2 is adopted in region-1 and a mesh size ‘1.0’ is specified for the points on the curved boundary of region-2. For all interior points of region-2, a linear interpolation is used to define their corresponding mesh sizes. And finally, in the remaining domain, a mesh size ‘2.0’ is used. The result obtained is plotted together with the fine-case of current resolution (see previous section) in the Figure A.16, while corresponding RMSE along with the RMSE of previous selected case is given in the Table A.14. Figure A.16 reveals that again the result obtained is very close to the one on the fine mesh. Moreover, Table A.14 shows that the current case leads to around a 50% reduction in the number of nodes without significantly affecting the RMSE.

A.4.3 Resolution 3

In this test, resolution 3 is adopted in region-1 and a mesh size ‘0.7’ is specified for the points on the boundary of region-2. For all interior points of region-2, a linear interpolation is used to define their corresponding mesh sizes. Like-wise previous section, a region R of dimensions $[-20, 20] \times [-20, 20] \times [-40, 0]$ is defined outside of region-2 and a mesh size ‘1.0’ is specified for the points in the region $R-$region-2, while in the remaining domain a mesh size ‘2.0’ is used. The result obtained is plotted together with fine-case of current resolution (see previous section) in the Figure A.17, while corresponding RMSE along with the RMSE of previous selected case is given in the Table A.15. Figure A.17 reveals that again the result obtained is very close to the one on the fine mesh. Moreover, Table A.15 shows that the current case leads to around a 50% reduction in the number of nodes without significantly affecting the RMSE.



(a)

(b) Zoom-in

Figure A.17: A comparison of elastic deformation profiles computed on the selected (meshing strategy 2) and fine mesh of resolution 3.

A.5 EHL

The mesh analysis tests done so far were by using the Hertzian pressure profile only. The Hertzian pressure profile defines the pressure values within a unit circular disc i.e outside this region pressure values are zeros. This was the reason behind choosing the dimension of central most region as unit cube/cuboid or unit semi-sphere. The actual EHL pressure profile is different than the Hertzian pressure profile i.e. for a typical EHL problem, pressure is not zero outside the Hertzian contact region, however, this is very small. For a sufficiently accurate elastic deformation solution for a typical EHL pressure profile, we need to choose a slightly bigger inner most region e.g. region-1 should be

Table A.16: Set of parameters for the contact between steel surfaces

Parameters	Values
Moes parameter, L	10
Moes parameter, M	20
Maximum Hertzian pressure, p_h	0.45GPa
Viscosity index, α	2.2×10^{-8}Pa^{-1}
Viscosity at ambient pressure, η_0	0.04 Pa s

Table A.17: RMSE of solutions, on current selected mesh (meshing strategy 2) and the fine mesh of resolution 1, with respect to the reference solution.

Case	nodes	RMSE
Fine-case	3505403	1.64×10^{-3}
Selected-case	97687	2.43×10^{-3}

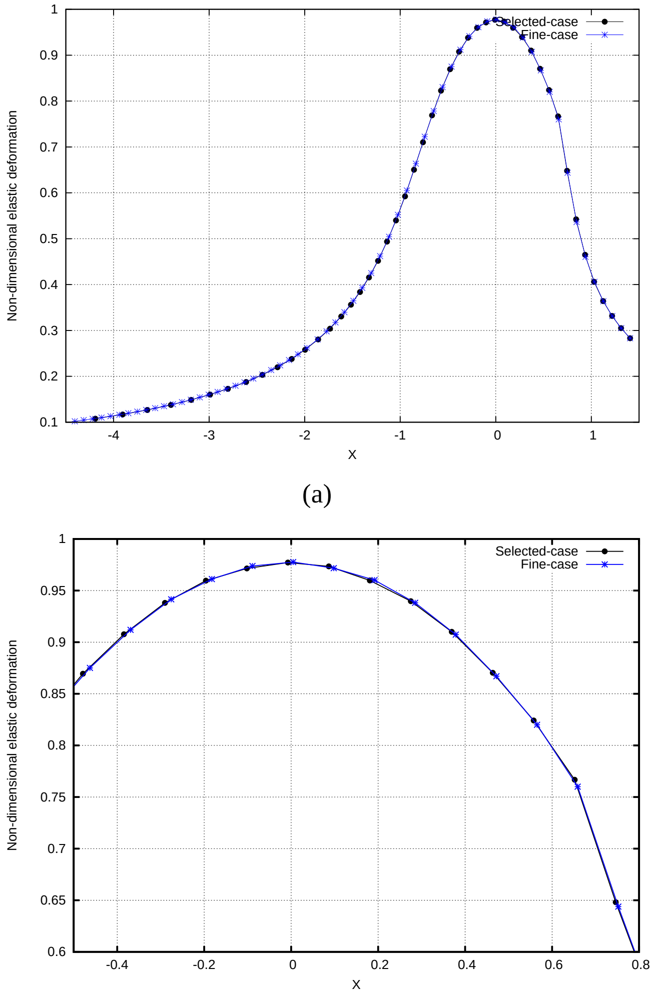
of radius at least ‘1.5’ because this is the region from where the pressure values start becoming significant, especially in the inlet region. Therefore, for all the experiments in this section the meshes use the same strategy as given in section A.4 however now the central most region is chosen to be of radius ‘1.5’.

It has been shown by Habchi [50, 51] that for heavily loaded cases, in addition to stabilization technique (see Chapter 4), an artificial diffusion is required to get a completely smooth pressure profile. Since, contrary to stabilization techniques, artificial diffusion is non-residual dependent, it has a small effect on the accuracy of the solution. Therefore, for a quantitative analysis, a relatively light loaded case is chosen in order to get a smooth pressure profile without the use of artificial diffusion. The EHL test case considered in this work is given in Table A.16.

It should be noted that this EHL test case is solved over the meshes used in section A.2, of which again the extreme case of resolution 3 will be used as a reference solution. In the following sub-sections, a selected case is defined for each resolution and the RMSE of this selected case along with its fine case will be calculated with respect to this reference solution.

A.5.1 Resolution 1

In this test, resolution 1 is adopted in the central region which is circular disc of radius ‘1.5’ and a mesh size ‘1.0’ is specified for the points on the curved boundary of semi-spherical region of radius ‘15’. For all interior points of region-2, a linear interpolation is used to define the corresponding mesh sizes. And finally, in the remaining domain, a mesh size ‘4.0’ is used. The result obtained is plotted together with the fine-case of the current resolution in the Figure A.18, while the corresponding RMSE, along with the RMSE of



(a)

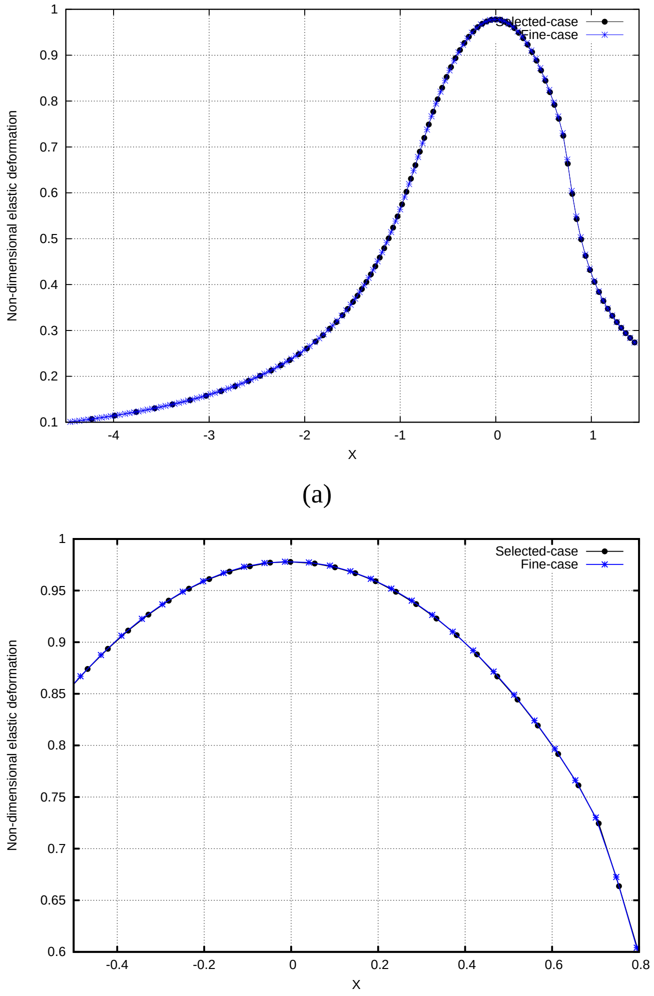
(b) Zoom-in

Figure A.18: A comparison of elastic deformation profiles computed on the selected (meshing strategy 2) and fine mesh of resolution 1.

finest case of the current resolution, is given in the Table A.17. Figure A.18 reveals that the result obtained is very close to the one using a very fine mesh. Moreover, Table A.17 shows that selected mesh leads to very small number of nodes (97687 nodes) without significantly affecting the RMSE (obtained with the use of a very fine mesh (3505403 nodes)).

A.5.2 Resolution 2

In this test, resolution 2 is adopted in the central region which is circular disc of radius ‘1.5’ and a mesh size ‘1.0’ is specified for the points on the curved boundary of semi-



(a)

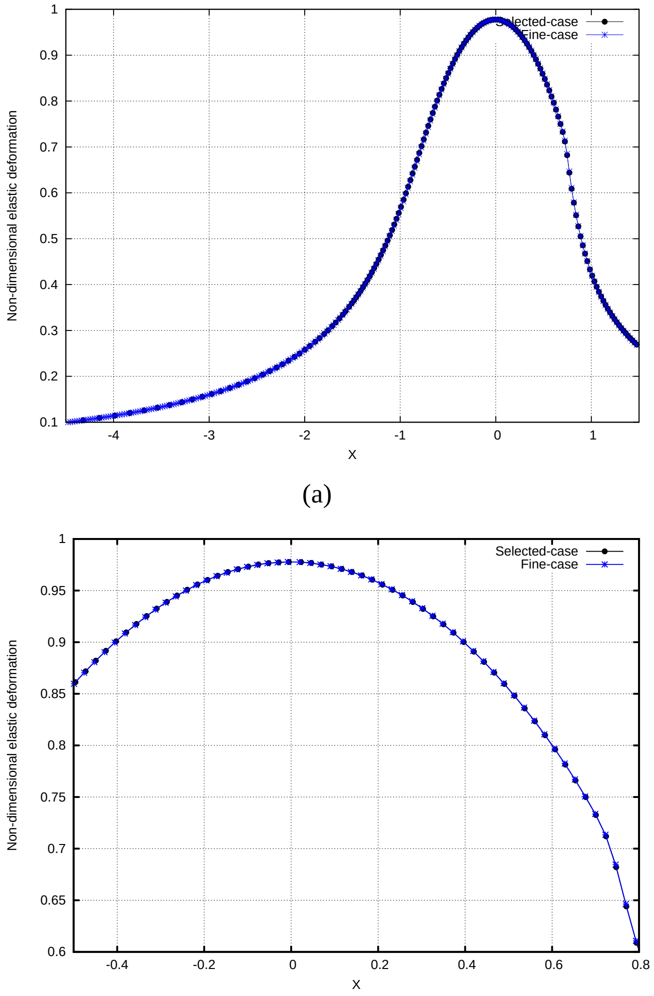
(b) Zoom-in

Figure A.19: A comparison of elastic deformation profiles computed on the selected (meshing strategy 2) and fine mesh of resolution 2.

Table A.18: RMSE of solutions, on current selected mesh (meshing strategy 2) and the fine mesh of resolution 2, with respect to the reference solution.

Case	nodes	RMSE
Fine-case	4950405	6.25×10^{-4}
Selected-case	221260	9.85×10^{-4}

spherical region of radius ‘15’. For all interior points of region-2, a linear interpolation is used to define the corresponding mesh sizes. And finally, in the remaining domain, a mesh size ‘2.0’ is used. The result obtained is plotted together with the fine-case of current resolution in the Figure A.19, while the corresponding RMSE, along with the RMSE of finest case of the current resolution, is given in the Table A.18. Figure A.19 reveals that the



(a)

(b) Zoom-in

Figure A.20: A comparison of elastic deformation profiles computed on the selected (meshing strategy 2) and fine mesh of resolution 3.

result obtained is very close to the one with using a very fine mesh. Moreover, Table A.18 shows that selected mesh leads to very small number of nodes (221260 nodes) without significantly affecting the RMSE (obtained with the use of a very fine mesh (4950405 nodes)).

A.5.3 Resolution 3

In this test, resolution 3 is adopted in the central region which is circular disc of radius ‘1.5’ and a mesh size ‘0.7’ is specified for the points on the curved boundary of semi-spherical region (region-2) of radius ‘15’. For all interior points of region-2, a linear

Table A.19: RMSE of solutions, on current selected mesh (meshing strategy 2) and the fine mesh of resolution 3, with respect to the reference solution.

Case	nodes	RMSE
Fine-case	15802299	0.0×10^0
Selected-case	705860	4.80×10^{-4}

interpolation is used to define the corresponding mesh sizes. As for the previous section, a region R of dimensions $[-20, 20] \times [-20, 20] \times [-40, 0]$ is defined outside of region-2 and a mesh size '1.0' is specified for the points in the region $R-$region-2, while in the remaining domain a mesh size '2.0' is used. The result obtained is plotted together with the fine-case of current resolution in the Figure A.20, while the corresponding RMSE, along with the RMSE of the finest case of current resolution, is given in the Table A.19. Figure A.20 reveals that the result obtained is very close to the one with using a very fine mesh. Moreover, Table A.19 shows that selected mesh leads to very small number of nodes (705860 nodes) without significantly affecting the RMSE (obtained with the use of a very fine mesh (15802299 nodes)).

A.6 Conclusion

Throughout this appendix, we have addressed a number of experiments to define the most appropriate mesh sizes needed throughout the domain in order to get a sufficiently accurate elastic deformation solution. The two different meshing strategies have been used to generate meshes. In the first strategy, the domain was split into a number of regions and an appropriate mesh size was adopted everywhere in each region. It was observed that splitting a region into further sub-regions and varying the mesh sizes between these new sub-regions leads to a mesh with fewer nodes without significantly affecting the accuracy of solution. However, defining more and more sub-regions along with appropriate mesh sizes is not always a simple job. Therefore, another strategy was used in which a mesh size corresponding to a certain resolution was adopted in the central region and then this mesh size was gradually increased while moving away from central region. This strategy leads to a greater decrease in the number of nodes in the mesh without having a significant effect on the accuracy of solution.

Initially, a Hertzian pressure profile was used to undertake all the analysis tests and later on the work was extended by considering a typical EHL pressure profile (see section A.5). It has been shown in section A.5 that the solution obtained using the selected meshes was sufficiently accurate but with a considerably lower computational cost. It should be noted that, throughout this analysis linear finite elements have been used to solve the linear elasticity equation and the EHL problem. The results of this appendix have been used in the work repeated in Chapter 6: ensuring that the tests of the efficiency of the proposed preconditioner have been undertaken on the most appropriate computational grids.