Adaptive Numerical Simulations for a Moving Boundary Problem Related to Pitting Corrosion

by

© **Abu Naser Sarker**

A thesis submitted to the School of Graduate Studies
in partial fulfilment of the requirements for
the degree of **Doctor of Philosophy**
in Scientific Computing.

Scientific Computing
MEMORIAL UNIVERSITY OF NEWFOUNDLAND

Spring 2023

St. John's, Newfoundland and Labrador, Canada

This is to certify that the thesis written by **Abu Naser Sarker** entitled: Adaptive Numerical Simulations for a Moving Boundary Problem Related to Pitting Corrosion has been approved by his committee as satisfactory completion of the thesis requirement for the degree of Doctor of Philosophy in Scientific Computing.

Supervised by

Dr. Ronald D. Haynes, Associate Dean of Graduate Studies
Professor, Department of Mathematics and Statistics, Memorial University

Dr. Michael Robertson
Professor, Department of Physics, Acadia University

Date

MEMORIAL UNIVERSITY OF NEWFOUNDLAND

St. John's, Newfoundland and Labrador, Canada.

Abstract

Many real-world science and engineering problems are described mathematically by partial differential equations (PDEs). Most of these PDEs can not be solved analytically and instead their solutions must be approximated on a computer. One numerical method capable of high accuracy solutions is the finite element method where the physical system is discretized using set of elements called a mesh. These elements consist of nodes or points, and the PDE is solved at each node of the mesh. We need high-quality meshes in order to achieve accurate numerical solutions to PDEs and an adaptive mesh that moves as the system evolves has many desirable properties. Moving meshes are now widely used in the numerical solution of PDEs, especially when dealing with problems that involve significant changes in the solution, such as fast-moving fronts, or moving boundary problems. A nonuniform mesh can maintain accuracy and also boost the efficiency of existing methods by automatically adjusting to solution behaviour and concentrating mesh points in critical areas, while minimizing the number of mesh nodes.

The main focus of this thesis is the design and implementation of an adaptive moving mesh method for a moving boundary problem related to pitting corrosion with homogeneous and heterogeneous materials. Pitting corrosion is one of the most devastating localized forms of corrosion generating a small pit, cavity or hole in the metal. Damage due to pitting corrosion of metals cost governments and industry billions of dollars per year and can put human lives at risk.

The **first part** of this research develops an adaptive moving mesh method for simulating pitting corrosion. The adaptive mesh is generated automatically by solving a mesh PDE coupled to the pitting corrosion PDE model. The moving mesh approach is shown to enable initial mesh generation, provide mesh recovery, and is able to smoothly tackle changing pit geometry. Materials with varying crystallography are considered as are single and multiple pits. A procedure is presented which allows pits to merge without a change in mesh topology, allowing computation to proceed without restarting. We have presented a robust, fully automatic, moving mesh solution framework for pitting corrosion.

The **second part** of this research is aimed at developing an adaptive moving mesh method for simulating pitting corrosion in materials containing heterogeneous inclusions. Inclusions are regions of a material that have a different composition or properties than the surrounding material. This makes for a challenging task due to the presence of the inclusion-type domains. In order to handle moving boundary domains with an inclusion, the metric is modified according to the location of the inclusions. The moving mesh approach using r-refinement is shown to handle changing pit geometry, including materials with varying crystallography, corrosion-resistant inclusions, and material voids.

r-refinement alone was not able to provide high mesh density near the inclusion(s) for long simulation times due to the obstacle(s) and the moving front. To overcome this issue, we propose a combination of h- and r- refinement, which is the focus of the **third part** of the research. h-refinement adds mesh elements by dividing each existing element into two or more elements and maintaining the type of element used. We design and implement an adaptive hr-refinement procedure for the simulation of pitting corrosion with heterogeneous materials. The adaptive hr-refinement is demonstrated to handle changing pit geometry, including materials with varying crystallography and corrosion-resistant inclusions.

The three main components of the research include theory, modeling, and application, which aim to provide effective and efficient meshes over complex moving domains in the solution of the pitting corrosion problem. The research is also focused on the development of software providing a new extension of MMPDElab with hr–refinement.

Lay Summary

Many mathematical models that involve PDEs cannot be solved analytically. To overcome this, numerical methods are used to approximate the solutions. A popular approach is to discretize the physical system and solve the resulting discrete version of the problem. An effective method is the finite element method, where the physical system is divided into smaller elements called a mesh. These elements consist of nodes or points, and the model problem is solved at each node of the mesh. In order to achieve accurate numerical solutions for the model problem, we need high-quality meshes.

The goal of this research work is to design and implement an adaptive moving mesh method for a moving boundary problem related to pitting corrosion. Pitting corrosion is localized corrosion; it occurs on metal surfaces and creates small holes, cavities or pits in the metal. Additionally, leaks can be caused by pitting that penetrates too deeply into the metal leading to further physical damage and degradation of the metal. Accurately detecting pit shapes is challenging due to the complex nature of pitting corrosion. The shapes of the pits depend on many factors, such as the components of the metal, the surface orientation, and the physical and chemical environment during the corrosion process. The question arises: How can predict pit shapes accurately? One approach is that a set of nodes with connected edges can detect the pit shapes easily. Obtaining pit shapes more accurately requires increasing the number of nodes in the regions around the pits. As time progress, the pit evolves, and additional nodes are needed to represent the actual pit shapes. Therefore, to precisely predict the pit shape over time, we need to adjust the position of the nodes near the pit boundary or add extra points on the pits. This technique is known as *mesh adaptation*.

In the first phase of the thesis, we develop an adaptive moving mesh framework for pitting corrosion with single and multiple pits. An adaptive moving mesh method is a powerful technique to achieve a more accurate solution. Based on the pitting corrosion mathematical model, we have shown that moving mesh methods automatically redistribute mesh nodes to increase the nodes around the pit(s) area as time progresses. This technique is known as r-refinement. During the corrosion process, the material crystallography considered. In addition, a procedure is presented which allows pits to merge without a change in mesh topology. We have presented a robust, fully automatic, moving mesh solution framework for pitting corrosion.

In the second phase of the thesis, we develop an adaptive moving mesh framework for pitting corrosion, involving heterogeneous materials. Inclusion type domains are regions of a material that have a different composition than the surrounding material. We have demonstrated that our technique can smoothly tackle the changing pit geometry associated with materials with inclusions of varying crystallography, corrosion-resistant inclusions, and material voids.

Finally, in the third phase of the thesis, we introduce a technique that can redistribute the mesh nodes around the pits and add extra nodes around the pits when required—this technique is known as hr-refinement. We show that adaptive hr-refinement can handle changing pit geometry better, including materials with varying crystallography and corrosion-resistant inclusions.

The research is also focused on developing software, specifically a new extension of MMPDElab providing hr–refinement. The main focus of this thesis is the design and implementation of an adaptive moving mesh method for a moving boundary problem related to pitting corrosion with homogeneous and heterogeneous materials.

Dedicated to my parents

Noor Mohammad Sarker and **Mst. Hasema Begum**

who made all impossibilities possible for me. Also, dedicated to my beloved siblings, respected teachers, closest friend, and reliable confidant.

Acknowledgements

First of all, I would like to express my deepest gratitude to my primary supervisor Dr. Ronald D. Haynes for his guidance, consistent support, and valuable advice throughout my PhD journey. Especially, his encouragement has greatly helped me in pursuing research work in scientific computing. At the same time, his guidance and teaching have enabled me to grasp the essential concepts relevant to my research. His contribution throughout all the steps of my thesis work is beyond expression. I deeply appreciate his intellectual insights and skills, as demonstrated by the excellence of the present work.

I am immensely grateful to my co-supervisor, Dr. Michael Robertson from Acadia University, for his valuable discussions, advice and guidance throughout this thesis work. In particular, his discussions on various areas in material science during the group meetings expanded my knowledge in corrosion and material science. His invaluable ideas and suggestions played a vital role in the success of this research. He has also played a significant role in enhancing the quality of graphs and publications. Additionally, his vast expertise in his research field consistently inspired me to learn new things and acquire knowledge.

I sincerely thank other committee members and research group members: Dr. Scott MacLachlan, Dr. Alex Bihlo, Dr. Jahrul Alam, Dr. Ivan Booth and Dr. James Munroe. From their fantastic lectures, I learned a lot which greatly contributed to my confidence in my research activities. Especially, I have acquired coding skills in C and utilizing the OpenMP and MPI libraries through discussions with Dr. Munroe for the project on *Algorithms for Distributed and Shared Memory Computers*. I have developed a clear understanding of numerical methods and numerical analysis through Dr. MacLachlan's lectures on *Iterative Methods in Numerical Linear Algebra*.

I would also like to thank all member of our research group of "Numerical Analysis and Scientific Computing" at Memorial University. The discussions during our the research group meetings, along with their innovative ideas and research-problem solving approaches, have enhanced my knowledge. Particularly, I would like to acknowledge my group mates, including Dr. Abdalaziz Hamdan and Dr. Rudige Brecht for their cooperation and support throughout my graduate study. I would like to extend my thanks to my colleagues and friends who have been along with me for the academic journey. Especially, I would like to mention Dr. Azim Mohammad, Mr. Rasel Biswas, Mr. Mohammad Arif, Mr. Tanzir Ahmed, and Mr. Zahangir Hossain for their mental support, valuable time and constant inspiration in helping me achieve my aspirations.

I would like to extend my thanks to Nancy Bishop, the interdisciplinary programs assistant. She has always been willing to do something to assist and support the students in her department. Sincere thanks to all the members of the other research groups, faculty, staff, and fellow students for their invaluable advice and dedication in successfully completing the thesis work. They are wonderful individuals and my well-wishers.

Special thanks to the School of Graduate Studies and the Department of Mathematics & Statistics for providing financial support throughout my studies. I am thankful to the department of Mathematics and Statistics for arranging the Teaching Assistantships for me. Additionally, I am grateful to the Memorial University of Newfoundland for their high performance computing support.

Finally, I would like to extend my deepest gratitude to my parents Noor Mohammad Sarker and Mst. Hasema Begum and my siblings for their unwavering love, consistent support and continuous motivation to reach my goals, especially during the most challenging moments of my life. They consistently stand by me in whatever I choose to pursue. I am eternally grateful to them for giving me the opportunities and valuable experiences that have made me into the person who I am today.

"The best among you is the one who doesn't harm others with his tongue and hands. The most faithful of you are the best mannered." – Abū al-Qāsim Muḥammad (PBUH)

Statement of Contributions

This thesis is an integrated, article-based thesis that presents three numerical investigations aimed at understanding the numerical modelling of pitting corrosion with different crystallography. Our approach is based on the adaptive moving mesh method that solves the PDE model equation. This thesis consists of the following three peer-reviewed articles that are either currently under review or in preprint form. The details of the contributions of the respective authors are described below.

Articles included in the thesis as chapters and authorship declaration:

1. Sarker, A. N., Haynes, R. D., Robertson, M. D. (2023), *Moving mesh simulation of pitting corrosion*, Journal of Mathematics in Science and Industry, Accepted.

2. Sarker, A. N., Haynes, R. D., Robertson, M. D. (2023), *A moving mesh simulation for pitting corrosion of heterogeneous materials*, Journal of the Computer & Mathematics with Applications, Submitted.

3. Sarker, A. N., Haynes, R. D., Robertson, M. D. (2023), *An adaptive hr-refinement simulation for pitting corrosion of heterogeneous materials*, Preprint.

Table 1 outlines the contributions made by the co-authors of each article. My primary role includes the preparation of each manuscript, developing the methods/algorithms and code, and the analysis of the corresponding results. The research problems described in Chapters 3, 4 and 5 were designed by myself in consultation with Dr. Ronald Haynes and Dr. Michael Robertson. However, the research ideas for each paper were initially planned by my supervisor, Dr. Ronald Haynes. I have adapted the assigned research questions to my background, knowledge, and interests. Therefore, the investigations and the research results contained in each paper were led by myself, while my supervisor edited each manuscript as necessary, and crosschecked all computer codes. Memorial University is the high-performance research computing facility provider for this research. The contributions of the authors are summarized in Table 1.

Table 1: The contributions of co-authors as indicated by author's initials: Abu Naser Sarker (ANS), Dr. Ronald D. Haynes (RH) and Dr. Michael Robertson (MR).

	Research proposal	**Code development**	**Perform simulations**	**Manuscript preparation**	**Contribution (in percent)**	**Status or comment**
Chapter 1				ANS	ANS (100%)	Introductory Chapter
Chapter 2				ANS	ANS (100%)	Background Chapter
Chapter 3	ANS, RH & MR	ANS & RH	ANS	ANS, RH & MR	ANS (80%) other (20%)	Accepted
Chapter 4	ANS, RH & MR	ANS & RH	ANS	ANS, RH & MR	ANS (80%) other (20%)	Submitted
Chapter 5	ANS, RH & MR	ANS & RH	ANS	ANS, RH & MR	ANS (80%) other (20%)	Preprint
Chapter 6				ANS	ANS (100%)	Conclusion Chapter

This work is considered of significant importance to the corrosion and materials science industries as it provides computational algorithms and tools for obtaining the corrosion pit behavior as a complementary technique to the challenging experimental work.

*The manuscripts presented in this thesis are slightly different than the published version due to formatting.

"No two things have been combined better than knowledge and patience. Seek knowledge from cradle to the grave." – Abū al-Qāsim Muḥammad (PBUH)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A wide variety of real-world science and engineering problems are challenging to examine directly, and experimentation in a laboratory can be an expensive and time consuming task. Computer simulation is indispensable for investigating these types of problems. Nowadays, computer simulations are widely used in various fields, such as meteorology, nuclear physics, biomedical, medical sectors, etc., and are expected to be utilized in nearly all scientific and engineering disciplines eventually. Thus, scientific simulation is becoming an integral part of the study of both real-world and theoretical problems. The procedure involves constructing a mathematical representation of the real-world problem to replicate its dynamic actions accurately. Many representations of problems are described by a system of differential equations (DEs) or partial differential equations (PDEs). The mathematical representation is known as a *mathematical model* of the problem.

1.1 Motivation

Mathematical models are highly useful tools for comprehending and analyzing the characteristics of a system. The model equations describing real-world problems can often be difficult to solve analytically. To overcome this difficulty, various sophisticated numerical methods have been developed over the past few decades, which allow the calculation of an approximate solution using a computer. Methods like finite differences, finite elements or finite volumes can estimate the solution value at specific points, known as *mesh* or *grid* points. Structured grids consisting of quadrilaterals (in 2D) and hexahedral brick elements (in 3D) are not suitable for discretizing complex geometries. The finite difference method (FDM) relies on structured grids, as a result the FDM is not an ideal choice for simulations of such systems. Instead, unstructured meshes composed of triangles in 2D and tetrahedra in 3D are more effective in discretizing arbitrary complex geometries. The finite element method (FEM) allows the use of unstructured meshes and has thus become a popular choice for performing simulations in various scientific computing fields.

Obtaining accurate numerical solutions requires a high quality mesh tailored to the specific model problem. Creating meshes for complex 2D or 3D geometries can be computationally expensive if we require several hundred thousand or even millions of grid points. Using a uniform mesh across the complex geometry can result in a significant computational cost, particularly in multidimensional scenarios where the number of required mesh points may be prohibitively high. Additionally, it can be extremely challenging, if not impossible, to design a mesh that can accurately capture the characteristics of the physical phenomenon (such as pitting corrosion) without prior knowledge. A alternative technique is to concentrate a greater number of mesh points in regions of significant solution variation and fewer points in the rest of the domain. By adapting the mesh in this way, the total number of required mesh points is much lower than with a uniform mesh, which may result in cost savings. Adapting the mesh for computational simulations becomes extremely challenging due to the difficulty in identifying regions of significant variation in the solution and the need to consider complex geometries with moving boundaries. As a result, this challenge has given rise to considerable research on the topic of *adaptive meshing* or *adaptive mesh refinement*.

Adaptive meshing is a powerful technique that provides accurate and economical numerical solutions. The method systematically adds more nodes in regions of large error, which helps to ensure that the solution converges to the desired level of accuracy. This approach can be cost-effective because it attempts to maximize solution accuracy for a given computational budget [47, 53, 85]. The accuracy of an adaptive technique is limited by how accurately the physical phenomenon is mathematically represented. For example, modelling pitting corrosion requires a mesh that effectively captures the movement of boundaries and inclusion regions, while also considering material properties. The extent to which adaptive meshing can improve solution accuracy depends on the accuracy of the underlying pitting corrosion model.

Pitting corrosion, our application of interest, is a type of localized corrosion that causes small pits, cavities, or holes in the metal, and it is considered to be very destructive. It can be more hazardous than uniform corrosion since it is challenging to identify, anticipate, and hence prepare for. Furthermore, the pits are often covered with corrosion products that make them difficult to detect. Even a small and narrow pit, which may not result in significant metal loss, could cause a system or metal to fail [84]. Mathematical models have been used to represent the problem of pitting corrosion, including time-dependent PDEs with complex geometry that involve the movement of boundaries and/or inclusions. This makes the problem extremely difficult to adapt the mesh at the moving boundary. In addition, the MMPDElab package has been extended to handle this type of problem, where MMPDElab is a general adaptive moving mesh finite element solver for time dependent PDEs.

1.2 Literature review

1.2.1 Moving mesh methods

Moving mesh methods are now widely used numerical techniques for solving PDEs efficiently and accurately [55]. When dealing with problems that involve substantial variations in the solution such as rapidly moving fronts, generating an efficient mesh can be challenging. However, using non-uniform meshes can help maintain accuracy and improve the efficiency of current methods. These meshes automatically adjust to solution behavior, allowing for the concentration of mesh points in critical areas.

One of the well-known moving mesh methods is the Arbitrary Lagrangian-Eulerian (ALE) method, which was first introduced by Hughes [31]. In the ALE method, the computational domain is allowed to move relative to a fixed reference frame in which the computational mesh deforms. The governing equations are written in a moving reference frame and the mesh points are adjusted to maintain higher accuracy [31, 52]. One downside of the ALE method is that it may experience mesh tangling and distortion, especially when the boundary motion is fast or complex. This can cause numerical instabilities and inaccuracies.

Another well-known moving mesh approach is adaptive moving mesh [29]. Adaptive moving mesh methods systematically adds more nodes in regions of large error, which helps to ensure more accurate and efficient numerical solutions as well as the solution converges to the desired level of accuracy. Adaptive moving mesh methods can automatically provide a continuously varying mesh by controlling the size, shape and orientation of the mesh elements in the domain. There are three general adaptive strategies: h-refinement, which involves adding or removing nodes to an existing mesh to improve local grid resolution [3, 9, 54]; p-refinement, which utilizes higher order schemes to enhance local accuracy when the solution is relatively smooth [15, 20]; and r-refinement, where a fixed number of nodes are moved or relocated with the solution [5, 29, 54]. These approaches can be applied separately or in combination to acquire the desired result. The redistribution or r-refinement method can provide isotropic or anisotropic meshes by changing the functional which is minimized [29]. The equidistribution principle [12], which is a core principle of moving mesh methods, is typically utilized to implement mesh movement. A more in-depth explanation of moving mesh methods will be presented in Chapter 2.

Mesh movement approaches are divided into two categories: the velocity-based approach and the location-based approach [5, 8, 30]. Most of the velocity-based approaches are motivated by the Lagrangian algorithm, where the computational mesh flow is tightly associated with the fluid or material particles during movement. The Eulerian approach has a fixed computational mesh and the continuum moves respect to mesh nodes. The Eulerian and Lagrangian algorithms are commonly used in fluid dynamics and structural material problems, respectively [17]. In general, Eulerian meshes avoid mesh tangling and diffusive solutions, but the method can have difficulty adjusting to sharp material interfaces. One benefit of the Lagrangian approach is that the advective terms do not appear in the governing equations. Thus, the Lagrangian methods are less diffusive compared to that of the Eulerian method, while also maintaining sharp material interfaces [29]. The Arbitrary Lagrangian-Eulerian (ALE) methods are velocity-based methods, which are a combination of Lagrangian and Eulerian approaches [21, 22, 39, 50, 52, 76].

The main goal of the location-based mesh movement approach is to control the location of mesh points in particular locations. A typical choice of location-based mesh movement is the variational approach, which relocates the mesh points and mesh movement based on minimizing a functional, which is formulated to measure the difficulty or the error in the numerical solution [29]. A number of location-based algorithms have been developed based on variational approaches with others developed based on elliptic PDEs. Elliptic PDEs can be used to generate boundary-fitted meshes [69, 78], which is sometimes known as Winslow's approach [79]. Winslow's idea is generalized using a functional [4] that controls a combination of the mesh adaptivity, smoothness, and orthogonality conditions. A number of articles address the type of mesh adaptation functionals as based on mechanical models [32, 33, 34], vector fields [37], a weighted Jacobian matrix [38, 39], a matrix-valued diffusion coefficient [7, 26], and the equidistribution and isotropy (or alignment conditions) presented in [23]. The moving mesh PDE (MMPDE) method has been developed by several authors [6, 25, 25, 26, 27, 56], where the mesh movement is determined by a gradient flow equation and where the functional plays a vital role.

Hence, the adaptive moving mesh method redistributes the mesh in regions where the solution changes rapidly and keeps the mesh coarse in regions where the solution varies slowly. This results in redistribution based on error analysis of the solution, equidistribution and alignment conditions. To generate optimal meshes for moving boundary problems, an appropriate monitor function (or mesh density function) and proper values of the parameters of moving mesh PDEs are required. Finally, the adaptive mesh is generated by solving the moving mesh PDE [29]. A more in-depth explanation of moving mesh methods will be presented in Chapter 2.

1.2.2 Moving boundary problems

Moving mesh methods are widely used numerical techniques for solving PDEs with moving boundaries or interfaces according to the specific problem to be solved [81]. They have a diverse set of applications in many fields of study, including heat transfer, fluid dynamics, and chemical reaction engineering. Some interesting areas where moving mesh methods have been recently utilized include:

- The study of the mechanics of cell migration, tissue growth, and simulating blood flow in arteries and veins in *biomechanics* [51].

- The simulation of fluid flow problems with moving boundaries or interfaces in *computational fluid dynamics* [71].

- The geological simulation of groundwater flow and contaminant transport in porous media in *geosciences* [83],

- The growth of crystals and the behavior of materials under stress and deformation in *materials science* [67].

Therefore, moving mesh methods have a broad range of applications in various fields and they continue to be an active scientific area of research. In these problems, the boundaries of the physical domain can move or deform according to the problem phenomena, and it is necessary to adapt the mesh in every time to maintain accuracy of the solution [45, 68]. In recent years, moving mesh methods have been a popular choice for many researchers and engineers due to their ability to provide accurate solutions and their ability to efficiently capture the dynamics of moving interfaces.

Moving mesh methods have proven their ability to maintain the accuracy and efficiency for solving moving boundary problems [28, 81]. However, there are some challenges associated with moving mesh methods. For example, the choice of mesh adaptation algorithm or monitor function, the treatment of singularities, and the stability of the method [16, 46, 55]. In spite of these challenges, moving mesh methods have gained popularity and have become a necessary technique for solving moving boundary problems in a wide range of applications. In this thesis, we will apply the adaptive moving mesh method for the solving moving boundary problem related to pitting corrosion in homogeneous and heterogeneous materials.

1.2.3 Corrosion and Pitting corrosion

Corrosion is a deterioration of a metal surface due to chemical or electrochemical reactions with its surrounding environment. If the total area of the corrosive sites are smaller than the total surface area, then the metal is said to be undergoing localized corrosion. Pitting corrosion is one of the most disastrous and devastating localized forms of corrosion generating a small pit, cavity, or hole in the metal. Pitting corrosion is difficult to identify and can have a big impact on the structural integrity of metal [66, 84]. The geometries of the pits depend on many factors such as the components of the metal, the surface orientation, and the physical and chemical environment at the time of attack [62]. Corrosion pits can have different shapes [49] and with the ability to grow over time, failure of engineering structures such as bridges, pipelines, and nuclear power plants can result [10, 49, 57].

The three basic stages of pitting corrosion are the initiation stage, the metastable stage, and the stable stage. A pit is identified as stable if it is actively growing over time [1, 14]. The initiation of a pit is followed by metastable growth that leads to stable growth under the right balance between electrochemical and mass transport mechanisms. The initiation of the pit is relatively difficult to detect since it occurs rapidly and penetrates a metal without significant loss of weight. Pit initiation is a random phenomenon on a metal surface with most of the surface remaining unattacked. This makes pitting corrosion challenging to anticipate, detect, and prevent. Researchers have studied the pitting initiation behavior of different metals, including stainless steel [19, 43, 58], beryllium [19], pure aluminum, and aluminum alloys [36, 40, 61, 70, 82]. Davis [11] has discussed metastable pitting corrosion of aluminum with single crystals. Stable pit growth and a computational model for the pit growth is discussed in [14].

Computational modeling and simulations of pitting corrosion have proven beneficial for studying pits under a wide range of conditions and materials. Available models include partial differential equation (PDE) based models [35, 41], phase-field models [2, 48] and probabilistic models [65?]. The PDE based computational model for pitting corrosion is derived from physical and electrochemical laws. The model is a fully-coupled system of the electrolyte and solid domains. The complete system depends on the time-dependent distortion of the pit and the solution chemistry within the pit including electrochemical reactions and mass transfer of species. The electrolyte potential distribution and the concentration of species can be obtained by solving the governing mass transport equations and the growth rate of the pit can be found from Faraday's Law. A complete solution of the mass transport equations is divided into the process of diffusion, electron-migration and chemical reactions, with the additional complexity of a moving solid boundary. Recently, an effective PDE-based model has been proposed which investigates time-dependent pitting corrosion in the presence of fluid flow [75].

Over the past few years, several review papers have been published with a primary focus on PDE-based models for pitting corrosion based on finite element or finite volume methods [13, 14, 42, 44, 60, 63, 64, 73]. In 2019, an extensive overview of the mathematical models for pitting corrosion based on anodic reaction at the corrosion front, transportation of ions in the pits of the electrolyte domain, and pit growth over time was provided [35]. The COMSOL® software package is often used to solve the PDE in the electrolyte domain and the corrosion front movement is computed by the arbitrary Lagrangian-Eulerian (ALE) and level set methods [14, 41]. In other studies, a 2D PDE model is solved with the finite element method [18, 72] and the finite volume method [59, 60]. Pit growth is determined by finite element methods and a level set approach in [72], and using an extended finite element method (XFEM) and level set method in [18]. In 2020, an ALE method is implemented to move the mesh at the pit boundary and analyze the relationship between the corrosion behavior and the local corrosive environment within a single pit [74]. In most of the existing studies, the COMSOL® multi-physics software is used for solving the system of PDEs and pit movement is implemented in a separate MATLAB program [14, 35, 77, 80]. The implementation details of the movement of the pit and corner nodes have not been described in detail in many of these papers. In this study, the implementation of the pit movement, the corner movement, and the numerical methods are fully described.

Determining the pit behavior experimentally is time consuming, expensive, and physically difficult or impossible in many situations. However, numerical simulations are suitable for studying pits under a wide range of conditions within a reasonable time. In our simulations, the stable stage of pit growth is considered. To our knowledge, an adaptive moving mesh method, our method of choice, has not been implemented for the PDE-based modeling of stable pitting corrosion. Adaptive moving mesh approaches have demonstrated great success in many application problems providing fine control over the spatial adaptivity and recovering numerical solutions which recover properties of the continuous solution. This serves as strong motivation to apply a moving mesh approach to our application problem.

In our moving mesh method, the FEM is used for the spatial discretization, and a solver is built based upon the software package MMPDElab by Huang [24]. MMPDElab is a general adaptive moving mesh finite element solver for time dependent PDEs. An adaptive moving mesh is used to get sufficient mesh elements in regions of the pit using an alternating mesh and physical solution approach. Movement of the nodes in the pit and choosing an appropriate monitor function are challenging tasks. We present numerical results for the electrolyte potential, the evolution of the corrosion pit front, as well as pit-depth and width at different times for different crystal orientations, corrosion-resistant inclusions, and material voids.

1.3 Objectives

The purpose of this thesis to provide an in-depth examination of both the theoretical and practical aspects of mesh adaptivity, with a specific focus on its implementation for the time-dependent moving boundary problems related to pitting corrosion. The main objectives of this thesis are given below:

1. Demonstrate the effectiveness of the moving mesh method in recovering meshes and tackling the changing geometry for the pitting corrosion problem with single

and multiple pits. We will apply r-refinement and show this technique is capable of handling the moving boundary smoothly.

2. Illustrate the ability of the moving mesh method for moving boundaries on inclusion-type domains for pitting corrosion with heterogeneous materials. Inclusion type domains are regions of a material that have a different composition compared to the surrounding material. We wish to show that r-refinement is able to provide sufficient meshes near the inclusion(s) and the pit(s), as required.

3. Present an hr-refinement approach for pitting corrosion with heterogeneous materials. The hr-refinement approach is a combination of h-refinement and r-refinement. We wish to demonstrate that hr-refinement can significantly improve the accuracy and efficiency of the simulation compared to r-refinement alone in some situations.

1.4 Contributions of this thesis

The majority of this work primarily focuses on developing a robust, automatic, adaptive moving mesh framework for pitting corrosion. The work involves the implementation of adaptive algorithms for simulating corrosion processes, as well as improving the accuracy of numerical simulations based on partial differential equation (PDE) models of corrosion processes. In the first part of the research, we develop and implement adaptive algorithms for the moving boundary problem, which are related to pitting corrosion. The r-refinement method provides mesh recovery and is able to smoothly tackle changing pit geometry. This work is presented in Chapter 3. In the second part of the research, we study heterogeneous materials with an emphasis on inclusion type domains and this work is presented in Chapter 4. In our observation of r-refinement in the second part of our research, we found that the mesh density near the inclusion decreases due to an obstacle (or inclusion) in front of the moving part of the boundary. To overcome this situation, h-refinement can be used. The h-refinement method involves using a more refined mesh consisting of the same type of element. This method divides each existing element into two or more elements while maintaining the type of element used. Therefore, the obvious way to improve mesh quality for our problem is to use a combination of h- and r- refinements. Finally, in the third part of the research we implement hr-refinement for heterogeneous materials with an emphasis on inclusion type domains.

1.5 Thesis organization

An outline of the thesis spread over the six chapters is as follows. **Chapter 1** gives the objectives and scope of the thesis as well as a relevant literature survey. In **Chapter 2**, we provide background materials for this thesis, including an overview of moving mesh methods, the preliminaries of the mechanism of pitting corrosion, crystal orientation, and a model problem. In **Chapter 3** we present "*Moving Mesh Simulations of Pitting Corrosion*". In this article we discuss the PDE model of pitting corrosion, proof of concept for simulations demonstrating the effectiveness of the moving mesh method for pitting corrosion, and implementation of single and multiple crystal directions to the package. **Chapter 4** presents our article "*A moving mesh simulation for pitting corrosion of heterogeneous materials*". In this article we discuss a PDE model for pitting corrosion and present a procedure on how to handle different inclusions and voids in the computational geometry. In **Chapter 5** we present our article "*An adaptive hr-refinement simulation for pitting corrosion of heterogeneous materials*". In this article we give an overview of hr-refinement and its implementation in our moving boundary problem. The final chapter is **Chapter 6**, which includes some important comments and provides several useful conclusions of the present research work and future research directions.

Bibliography

[1] KV Akpanyung and RT Loto. Pitting corrosion evaluation: A review. In *Journal of Physics: Conference Series*, volume 1378, page 022088. IOP Publishing, 2019.

[2] Talha Qasim Ansari, Zhihua Xiao, Shenyang Hu, Yulan Li, Jing-Li Luo, and San-Qiang Shi. Phase-field model of pitting corrosion kinetics in metallic materials. *npj Computational Materials*, 4(1):38, 2018.

[3] Scott B Baden. *Structured adaptive mesh refinement (SAMR) grid methods*, volume 117. Springer Science & Business Media, 2000.

[4] Jeremiah U Brackbill and Jeff S Saltzman. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics*, 46(3):342–368, 1982.

[5] Chris J Budd, Weizhang Huang, and Robert D Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.

[6] Weiming Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM Journal on Numerical Analysis*, 43(1):19–40, 2005.

[7] Weiming Cao, Weizhang Huang, and Robert D Russell. A study of monitor functions for two-dimensional adaptive mesh generation. *SIAM Journal on Scientific Computing*, 20(6):1978–1994, 1999.

[8] Weiming Cao, Weizhang Huang, and Robert D Russell. Approaches for generating moving adaptive meshes: location versus velocity. *Applied Numerical Mathematics*, 47(2):121–138, 2003.

[9] Graham F Carey. *Computational grids: generations, adaptation & solution strategies*. CRC Press, 1997.

[10] François Cattant, Didier Crusset, and Damien Féron. Corrosion issues in nuclear industry today. *Materials Today*, 11(10):32–37, 2008.

[11] BW Davis, PJ Moran, and PM Natishan. Metastable pitting behavior of aluminum single crystals. *Corrosion Science*, 42(12):2187–2192, 2000.

[12] Carl de Boor. Good approximation by splines with variable knots. In *Spline Functions and Approximation Theory*, pages 57–72. Springer, 1973.

[13] Dennj De Meo and Erkan Oterkus. Finite element implementation of a peridynamic pitting corrosion damage model. *Ocean Engineering*, 135:76–83, 2017.

[14] Virginia G DeGiorgi, Nithyanand Kota, Alexis C Lewis, and Siddiq M Qidwai. Numerical modeling of pit growth in microstructure. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55850, pages 1–7. American Society of Mechanical Engineers, 2013.

[15] Karen D Devine and Joseph E Flaherty. Parallel adaptive hp-refinement techniques for conservation laws. *Applied Numerical Mathematics*, 20(4):367–386, 1996.

[16] Yana Di, Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh methods for singular problems on a sphere using perturbed harmonic mappings. *SIAM Journal on Scientific Computing*, 28(4):1490–1508, 2006.

[17] Jean Donea, Antonio Huerta, J-Ph Ponthot, and Antonio Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. *Encyclopedia of Computational Mechanics*, 2004.

[18] Ravindra Duddu. Numerical modeling of corrosion pit propagation using the combined extended finite element and level set method. *Computational Mechanics*, 54(3):613–627, 2014.

[19] MA Hill, JF Bingert, and RS Lillard. The relationship between crystallographic orientation and the passivity and breakdown of beryllium. In *194th Meeting of the Electrochemical Society*. 1998.

[20] Michael Hillman and Kuan-Chung Lin. Consistent weak forms for meshfree methods: Full realization of h-refinement, p-refinement, and a-refinement in strong-type essential boundary condition enforcement. *Computer Methods in Applied Mechanics and Engineering*, 373:113448, 2021.

[21] CW Hirt, AA Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.

[22] Cyrill W Hirt, Anthony A Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[23] Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *Journal of Computational Physics*, 174(2):903–924, 2001.

[24] Weizhang Huang. An introduction to MMPDElab. *ArXiv Preprint ArXiv:1904.05535*, 2019.

[25] Weizhang Huang, Yuhe Ren, and Robert D Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM Journal on Numerical Analysis*, 31(3):709–730, 1994.

[26] Weizhang Huang and Robert D Russell. A high dimensional moving mesh strategy. *Applied Numerical Mathematics*, 26(1-2):63–76, 1998.

[27] Weizhang Huang and Robert D Russell. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM Journal on Scientific Computing*, 20(3):998–1015, 1998.

[28] Weizhang Huang and Robert D Russell. Adaptive mesh movement—the mmpde approach and its applications. *Journal of Computational and Applied Mathematics*, 128(1-2):383–398, 2001.

[29] Weizhang Huang and Robert D Russell. *Adaptive Moving Mesh Methods*, volume 174. Springer Science & Business Media, 2010.

[30] Weizhang Huang, Robert D Russell, Weizhang Huang, and Robert D Russell. Velocity-based adaptive methods. *Adaptive Moving Mesh Methods*, pages 379–399, 2011.

[31] Thomas JR Hughes, Wing Kam Liu, and Thomas K Zimmermann. Lagrangian-eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329–349, 1981.

[32] Olivier-Pierre Jacquotte. A mechanical model for a new grid generation method in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 66(3):323–338, 1988.

[33] Olivier-Pierre Jacquotte. Generation, optimization and adaptation of multiblock grids around complex configurations in computational fluid dynamics. *International Journal for Numerical Methods in Engineering*, 34(2):443–454, 1992.

[34] Olivier-Pierre Jacquotte and Grégory Coussement. Structured mesh adaption: space accuracy and interpolation methods. *Computer Methods in Applied Mechanics and Engineering*, 101(1-3):397–432, 1992.

[35] Siavash Jafarzadeh, Ziguang Chen, and Florin Bobaru. Computational modeling of pitting corrosion. *Corrosion Reviews*, 37(5):419–439, 2019.

[36] SMLC Jain, MLC Lim, JL Hudson, and JR Scully. Spreading of intergranular corrosion on the surface of sensitized Al-4.4 mg alloys: A general finding. *Corrosion Science*, 59:136–147, 2012.

[37] Patrick M Knupp. Mesh generation using vector fields. *Journal of Computational Physics*, 119(1):142–148, 1995.

[38] Patrick M Knupp. Jacobian-weighted elliptic grid generation. *SIAM Journal on Scientific Computing*, 17(6):1475–1490, 1996.

[39] Patrick M Knupp, Len G Margolin, and Mikhail Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176(1):93–128, 2002.

[40] EV Koroleva, GE Thompson, P Skeldon, and B Noble. Crystallographic dissolution of high purity aluminium. In *Proceedings of the Royal Society of London A:*

Mathematical, Physical and Engineering Sciences, volume 463, pages 1729–1748. The Royal Society, 2007.

[41] Nithyanand Kota, Siddiq M Qidwai, Alexis C Lewis, and Virginia G DeGiorgi. Microstructure-based numerical modeling of pitting corrosion in 316 stainless steel. *ECS Transactions*, 50(31):155–164, 2013.

[42] D Krouse, N Laycock, and C Padovani. Modelling pitting corrosion of stainless steel in atmospheric exposures to chloride containing environments. *Corrosion Engineering, Science and Technology*, 49(6):521–528, 2014.

[43] B Ravi Kumar, Raghuvir Singh, Bhupeshwar Mahato, PK De, NR Bandyopadhyay, and DK Bhattacharya. Effect of texture on corrosion behavior of aisi 304l stainless steel. *Materials Characterization*, 54(2):141–147, 02 2005.

[44] Nicholas J Laycock, Donal P Krouse, Shaun C Hendy, and David E Williams. Computer simulation of pitting corrosion of stainless steels. *The Electrochemical Society Interface*, 23(4):65–71, 2014.

[45] Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *Journal of Computational Physics*, 170(2):562–588, 2001.

[46] Ruo Li, Tao Tang, and Pingwen Zhang. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *Journal of Computational Physics*, 177(2):365–393, 2002.

[47] Rainald Löhner. Mesh adaptation in fluid mechanics. *Engineering Fracture Mechanics*, 50(5-6):819–847, 1995.

[48] Weijie Mai, Soheil Soghrati, and Rudolph G Buchheit. A phase field model for simulating the pitting corrosion. *Corrosion Science*, 110:157–166, 2016.

[49] Abdel Salam H Makhlouf, Victor Herrera, and Edgar Muñoz. Corrosion and protection of the metallic structures in the petroleum industry due to corrosion

and the techniques for protection. In *Handbook of Materials Failure Analysis*, pages 107–122. Elsevier, 2018.

[50] Len G Margolin. Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):198–202, 1997.

[51] Ramiro Moreno, Franck Nicoud, Louis Veunac, and Hervé Rousseau. Non linear transformation field to build moving meshes for patient specific blood flow simulations. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics*. Egmond aan Zee, The Netherlands, September 5-8, 2006.

[52] Fabio Nobile and Luca Formaggia. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-West Journal of Numerical Mathematics*, 7:105–132, 1999.

[53] E Onate and G Bugeda. Mesh optimality criteria for adaptive finite element computations. *The Mathematics of Finite Elements and Applications*, pages 121–135, 1994.

[54] Tomasz Plewa, Timur Linde, V Gregory Weirs, et al. Adaptive mesh refinement-theory and applications. 2005.

[55] Joan Remski, Jingyan Zhang, and Qiang Du. On balanced moving mesh methods. *Journal of computational and applied mathematics*, 265:255–263, 2014.

[56] Yuhe Ren and Robert D Russell. Moving mesh techniques based upon equidistribution, and their stability. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1265–1286, 1992.

[57] Pierre R Roberge. *Corrosion Engineering*. McGraw-Hill Education, 2008.

[58] Akinori Sato, Kenzo Kon, Shigeo Tsujikawa, and Yoshihiro Hisamatsu. Effect of crystallographic orientation on dissolution behavior of stainless steels single crystal. *Materials Transactions, JIM*, 37(4):729–732, 1996.

[59] Stefan Scheiner and Christian Hellmich. Stable pitting corrosion of stainless steel as diffusion-controlled dissolution process with a sharp moving electrode boundary. *Corrosion Science*, 49(2):319–346, 2007.

[60] Stefan Scheiner and Christian Hellmich. Finite volume model for diffusion-and activation-controlled pitting corrosion of stainless steel. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2898–2910, 2009.

[61] Jong Hyun Seo, Jong-Ho Ryu, and Dong Nyung Lee. Formation of crystallographic etch pits during AC etching of aluminum. *Journal of the Electrochemical Society*, 150(9):B433–B438, 2003.

[62] SM Sharland. A review of the theoretical modelling of crevice and pitting corrosion. *Corrosion Science*, 27(3):289–323, 1987.

[63] SM Sharland. A mathematical model of crevice and pitting corrosion—II. the mathematical solution. *Corrosion Science*, 28(6):621–630, 1988.

[64] SM Sharland and PW Tasker. A mathematical model of crevice and pitting corrosion-I. The physical model. *Corrosion Science*, 28(6):603–620, 1988.

[65] Elahe Shekari, Faisal Khan, and Salim Ahmed. Probabilistic modeling of pitting corrosion in insulated components operating in offshore facilities. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 3(1):011003–011003, 2017.

[66] Laura B Simon, Mohammad Khobaib, Theodore E Matikas, CS Jeffcoate, and MS Donley. Influence of pitting corrosion on structural integrity of aluminum alloys. In *Nondestructive Evaluation of Aging Materials and Composites III*, volume 3585, pages 40–47. International Society for Optics and Photonics, 1999.

[67] M Sistaninia and M Niffenegger. Prediction of damage-growth based fatigue life of polycrystalline materials using a microstructural modeling approach. *International Journal of Fatigue*, 66:118–126, 2014.

[68] Tao Tang. Moving mesh methods for computational fluid dynamics. *Contemporary Mathematics*, 383(8):141–173, 2005.

[69] Joe F Thompson, Frank C Thames, and C Wayne Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, 15(3):299–319, 1974.

[70] Grainne M Treacy and Carmel B Breslin. Electrochemical studies on single-crystal aluminium surfaces. *Electrochimica Acta*, 43(12):1715–1720, 1998.

[71] Željko Tuković and Hrvoje Jasak. A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. *Computers & Fluids*, 55:70–84, 2012.

[72] A Turnbull, DA Horner, and BJ Connolly. Challenges in modelling the evolution of stress corrosion cracks from pits. *Engineering Fracture Mechanics*, 76(5):633–640, 2009.

[73] John C Walton. Mathematical modeling of mass transport and chemical reaction in crevice and pitting corrosion. *Corrosion Science*, 30(8-9):915–928, 1990.

[74] Kai Wang, Chenpei Li, Yanhui Li, Jinling Lu, Yueshe Wang, and Xingqi Luo. Multi-physics coupling analysis on the time-dependent localized corrosion behavior of carbon steel in CO_2-H_2O environment. *Journal of The Electrochemical Society*, 167(1):013505–013505, 2019.

[75] Kai Wang, Xiaobin Ma, Yueshe Wang, and Renyang He. Study on the time-dependent evolution of pitting corrosion in flowing environment. *Journal of The Electrochemical Society*, 164(7):C453–C463, 2017.

[76] BV Wells, Michael J Baines, and Paul Glaister. Generation of arbitrary Lagrangian–Eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations. *International Journal for Numerical Methods in Fluids*, 47(10-11):1375–1381, 2005.

[77] Joseph W Wilder, Curtis Clemons, Dmitry Golovaty, Kevin L Kreider, Gerald W Young, and R Scott Lillard. An adaptive level set approach for modeling damage due to galvanic corrosion. *Journal of Engineering Mathematics*, 91(1):121–142, 2015.

[78] Alan M Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.

[79] Alan M Winslow. Adaptive-mesh zoning by the equipotential method. Technical report, Lawrence Livermore National Lab., CA (USA), 1981.

[80] J Xiao and S Chaudhuri. Predictive modeling of localized corrosion: an application to aluminum alloys. *Electrochimica Acta*, 56(16):5630–5641, 2011.

[81] Xiangmin Xu. Moving mesh methods for moving boundary problems and higher order partial differential equations. 2008.

[82] M Yasuda, F Weinberg, and D Tromans. Pitting corrosion of Al and Al-Cu single crystals. *Journal of the Electrochemical Society*, 137(12):3708–3715, 1990.

[83] A Yekta, P Salinas, S Hajirezaie, MA Amooie, CC Pain, MD Jackson, C Jacquemyn, and MR Soltanian. Reactive transport modeling in heterogeneous porous media with dynamic mesh optimization. *Computational Geosciences*, 25:357–372, 2021.

[84] Xingliang Yu, Fangyi Wan, and Yingnan Guo. Micromechanics modeling of skin panel with pitting corrosion for aircraft structural health monitoring. In *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE, 2016.

[85] JZ Zhu and OC Zienkiewicz. Adaptive techniques in the finite element method. *Communications in Applied Numerical Methods*, 4(2):197–204, 1988.

Chapter 2

Background

In this chapter, we provide a broad introduction to the key foundational concepts necessary to understand the adaptive moving mesh methods as well as essential tools for our later work. We discuss the basic principles of the moving mesh method including the equidistribution and alignment conditions. Next, the formulation of the Moving Mesh Partial Differential Equations (MMPDEs), the governing equations that describe the motion of the mesh points is given. We also discuss the design of monitor functions or mesh density functions, which are used to guide the mesh adaptation process. To illustrate mesh adaptation, we introduce the standard moving mesh algorithm for the well-known one-dimensional Burgers' equation. We explain how the mesh is automatically adjusted at the proper location to adapt to the solution behavior. Finally, we will discuss the fundamentals of corrosion and pitting corrosion.

2.1 Moving mesh methods

Moving meshes have gained popularity in the numerical solution of partial differential equations (PDEs), as they improve the efficiency and accuracy of existing approximation techniques by automatically adapting to the solution behavior and concentrating mesh points in regions of interest. The moving mesh method automatically redistributes a fixed number of nodes where additional accuracy is required.

2.1.1 Equidistribution principle and MMPDE in 1D

The equidistribution principle (EP) plays a vital role in mesh adaptation. In 1D, the EP is used to derive the moving mesh PDE (MMPDE). The EP was introduced by deBoor [7], Dodson [10] and White [33]. It is based on very simple idea: if some measure of the error or mesh density function $\boldsymbol{\rho} = \boldsymbol{\rho}(x) > 0$ is given. A good choice of a mesh would be a mesh for which the error is equally distributed over all subintervals or the mesh elements in the domain. Given an integer $N > 1$, the continuous and bounded function $\boldsymbol{\rho}$ on $[a, b]$ is evenly distributed on the mesh $\mathcal{T}_h = a = x_0 < x_1 < ... < x_N = b$, if

$$\int_{x_0}^{x_1} \boldsymbol{\rho}(x)dx = \int_{x_1}^{x_2} \boldsymbol{\rho}(x)dx = ... = \int_{x_{N-1}}^{x_N} \boldsymbol{\rho}(x)dx. \tag{2.1}$$

This forces the area under $\boldsymbol{\rho}$ on all subintervals to be the same. A mesh $\mathcal{T}_h$ is called an equidistributing mesh if the mesh satisfies the EP. The function $\boldsymbol{\rho}$ is known as the mesh density function and the function $\boldsymbol{\rho}^2$ is called a *monitor function*.

Let us derive the mesh PDE using the EP for a general steady-state boundary value problem in one-dimension

$$\mathcal{L}\{u\} = 0, \quad u(a) = p, \quad u(b) = q, \tag{2.2}$$

where $\mathcal{L}$ is a spatial differential operator. A uniform mesh does not provide an accurate and efficient solution when the BVP has a“difficult” solution. The physical problem in the non-uniform x-coordinate is transformed to the computational uniform ξ-coordinate with domain $\Omega_c = [0, 1]$, where $x(0) = a$ and $x(1) = b$. Here $\Omega = [a, b]$ is known as the physical domain. We attempt to generate a mesh $\mathcal{T}_h$ using a mesh transformation $x = x(\xi) : \Omega_c \to \Omega$ and a uniform mesh in the ξ-coordinate

$$\xi_i = \frac{i}{N}, \quad i = 0, 1, ..., N.$$

Equation (2.1) can be written as

$$\begin{aligned}\int_a^{x_i} \boldsymbol{\rho}(x)dx &= \frac{i}{N}\int_a^b \boldsymbol{\rho}(x)dx \\ &= \frac{i}{N}\boldsymbol{\sigma}, \quad i = 0, 1, ..., N\end{aligned}$$

where $\boldsymbol{\sigma} = \int_a^b \boldsymbol{\rho}(x)dx$. The function $\int_a^x \boldsymbol{\rho}(x)dx$ is strictly monotonically increasing if $\boldsymbol{\rho} > 0$, and therefore each x_i is unique. Here $\boldsymbol{\sigma}$ and $\frac{i}{N}\boldsymbol{\sigma}$ are the total error and average error in the approximating solution, respectively. Essentially, if $\boldsymbol{\rho}$ is large where the error of the computed solution is large, then the EP will force the mesh points to be close to each other in that region. Using the mesh transformation, we have

$$\int_a^{x(\xi_i)} \boldsymbol{\rho}(\tilde{x})d\tilde{x} = \xi_i\boldsymbol{\sigma}, \quad i = 0, 1, ..., N.$$

The continuous version of this equation is

$$\int_a^{x(\xi)} \boldsymbol{\rho}(\tilde{x})d\tilde{x} = \xi\boldsymbol{\sigma}, \quad \forall\xi \in \Omega_c. \tag{2.3}$$

The continuous mapping $x = x(\xi)$ is called the *equidistributing coordinate transformation* if it satisfies relation (2.3). Differentiating with respect to ξ gives

$$\boldsymbol{\rho}(x)\frac{dx}{d\xi} = \boldsymbol{\sigma}. \tag{2.4}$$

Equation (2.4) indicates that $\frac{dx}{d\xi}$ is small when $\boldsymbol{\rho}$ is large. Again, differentiating with respect to ξ gives

$$\frac{d}{d\xi}\left(\boldsymbol{\rho}(x)\frac{dx}{d\xi}\right) = 0 \tag{2.5}$$

with the boundary conditions

$$x(0) = a, \quad x(1) = b. \tag{2.6}$$

This is a nonlinear boundary value problem for the required mesh transformation and physical mesh. The mesh is determined by solving the mesh equation (2.2) and the physical PDE of interest as a coupled system.

The equidistribution principle is used to appropriately manage the desired size of the mesh elements. In the multidimensional case, an equidistributing mesh can be represented as a uniform mesh in a metric space [17].

2.1.2 Equidistribution and alignment conditions in multiple dimensions

An adaptive mesh is generated as an image of a reference mesh using a coordinate transformation $x = x(\xi) : \Omega_c \to \Omega$, where Ω_c is the computational domain (equipped usually with a uniform mesh) and Ω is the physical domain (which will have a non-uniform mesh). The Jacobian matrix $\mathbf{J}$ of the coordinate transformation is calculated by $\mathbf{J}(\xi) = \frac{\partial x}{\partial \xi}(\xi)$. By linearizing a series expansion about a point ξ_0 in Ω_c, we have

$$x(\xi) = x(\xi_0) + \mathbf{J}(\xi_0)(\xi - \xi_0) + \mathcal{O}(|\xi - \xi_0|)^2.$$

This clearly shows that the Jacobian $\mathbf{J}(\xi_0)$ determines the behavior of the transformation around ξ_0. Since a uniform mesh is used on Ω_c, the Jacobian plays a vital role in the size, shape, and orientation around ξ_0. The matrix $\mathbf{J}(\xi_0)$ can be expressed by its singular-value decomposition (SVD)

$$\mathbf{J}(\xi_0) = U\Sigma V^T,$$

where U and V are orthogonal matrices, (i.e, $U^TU = UU^T = I$, $V^TV = VV^T = I$), and $\Sigma = \texttt{diag}(\sigma_1, \sigma_2, ..., \sigma_d)$, where σ_i's are the square roots of the eigenvalues of $\mathbf{J}(\xi_0)^T\mathbf{J}(\xi_0)$, or simply the singular values of $\mathbf{J}(\xi_0)$. Figure 2.1 shows a graphical illustration of the SVD of a unit ball. The shape and size are controlled by the singular values, Σ, and the orientation is controlled by U. Since the V matrix has no contribution in determining the size, shape, and orientation of the mesh, for mesh adaptation purposes we use the SVD of $\mathbf{J}^{-T}\mathbf{J}^{-1}$, written as

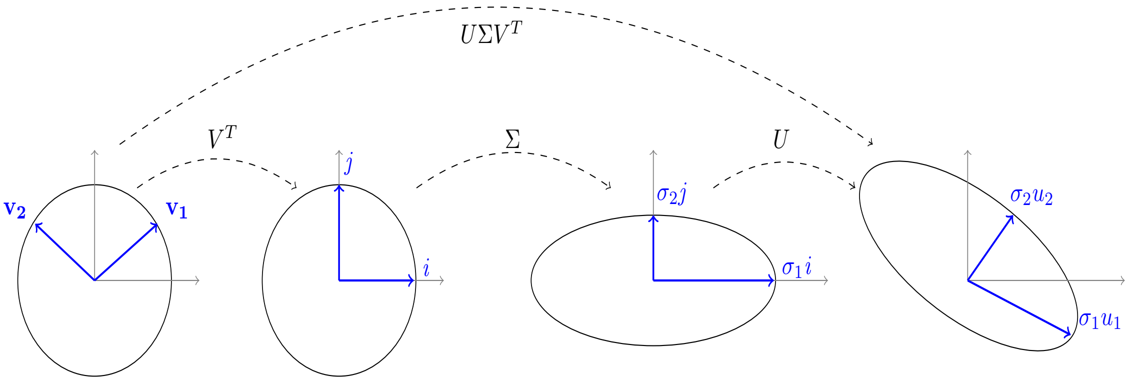


Figure 2.1: Graphical representation of SVD.

$$\mathbf{J}^{-T}\mathbf{J}^{-1} = U\Sigma^{-2}U^T.$$

Here, the columns of U are the eigenvectors and σ_i^{-2} are the eigenvalues of $\mathbf{J}^{-T}\mathbf{J}^{-1}$. So, the metric $\mathbf{J}^{-T}\mathbf{J}^{-1}$ plays a vital role in controlling the mesh elements. Thus, complete control of size, shape, and orientation of the mesh elements can be obtained by specifying the matrix $\mathbf{J}^{-T}\mathbf{J}^{-1}$. One choice is

$$\mathbf{J}^{-T}\mathbf{J}^{-1} = c^{-1}M(x) \tag{2.7}$$

where c is a constant and $M = M(x)$ is a user specified matrix-valued function, called a monitor-function. We assume M is a $d \times d$ symmetric and positive definite matrix. Let

$$\boldsymbol{\sigma} = \int_{\Omega} \boldsymbol{\rho}, \quad \boldsymbol{\rho} dx = \sqrt{\texttt{det}(M)}.$$

Rewriting equation (2.7) gives

$$\mathbf{J}^T M \mathbf{J} = cI, \quad \text{ in } \Omega_c. \tag{2.8}$$

To find the value of c, we take the determinant of both sides

$$\texttt{det}(\mathbf{J})^2 \texttt{det}(M) = \texttt{det}(cI).$$

Since $\boldsymbol{\rho}^2 = \texttt{det}(M)$, we have

$$\texttt{det}(\mathbf{J})^2 \boldsymbol{\rho}^2 = c^d.$$

This implies that

$$\texttt{det}(\mathbf{J}) \boldsymbol{\rho} = c^{\frac{d}{2}}.$$

Integrating over the computational domain Ω_c gives us

$$\begin{aligned} \int_{\Omega_c} \texttt{det}(\mathbf{J}) \boldsymbol{\rho} d\xi &= \int_{\Omega_c} c^{\frac{d}{2}} d\xi \\ &= c^{\frac{d}{2}} |\Omega_c|, \end{aligned}$$

where $|\Omega_c|$ is the size of the computational domain. Now, changing variables gives

$$\int_{\Omega} \boldsymbol{\rho} dx = c^{\frac{d}{2}} |\Omega_c|.$$

Hence, we have

$$c = \Big(\frac{\boldsymbol{\sigma}}{|\Omega_c|} \Big)^{\frac{2}{d}}.$$

Substituting the value of c in equation (2.8) we have

$$\mathbf{J}^T M\mathbf{J} = \Big(\frac{\boldsymbol{\sigma}}{|\Omega_c|}\Big)^{\frac{2}{d}} I. \tag{2.9}$$

The monitor function M determines the complete control of size, shape and orientation of the mesh elements in the whole domain if the coordinate transformation satisfies relation (2.9).

Denote $J = \texttt{det}(\mathbf{J})$ and take the determinant of both sides of the equation (2.9)

$$J^2\boldsymbol{\rho}^2 = \Big(\frac{\boldsymbol{\sigma}}{|\Omega_c|}\Big)^2.$$

This can be written as

$$J\boldsymbol{\rho} = \frac{\boldsymbol{\sigma}}{|\Omega_c|}. \tag{2.10}$$

This is a multi-dimensional generalization of the *equidistribution principle*. This implies that J (indicating the size of the mesh element) is large in the region of small $\boldsymbol{\rho}$ and is small in the region of large $\boldsymbol{\rho}$.

We observe that the eigenvalues of the matrix $\mathbf{J}^T M\mathbf{J}$ are the same from equation (2.9) and all eigenvalues, $\big(\frac{\boldsymbol{\sigma}}{|\Omega_c|}\big)^{\frac{2}{d}}$ are positive real numbers. The product of the eigenvalues is a constant, $\big(\frac{\boldsymbol{\sigma}}{|\Omega_c|}\big)^2$, which is the determinant of the matrix $\mathbf{J}^T M\mathbf{J}$. Hence, we obtain $(\boldsymbol{\rho}J)^2 = \texttt{det}(\mathbf{J}^T M\mathbf{J})$ from equation (2.9) and (2.10). In general, the arithmetic mean-geometric mean (AM-GM) inequality states that the arithmetic mean is greater than or equal to the geometric mean of n non-negative real numbers and the equality holds if the numbers are the same. Suppose λ_i, $i = 1, 2, ...d$, are the positive eigenvalues of the $\mathbf{J}^T M\mathbf{J}$. Thus, by the AM-GM equality we have

$$\Big(\prod_{i=1}^{d}\lambda_i\Big)^{\frac{1}{d}} = \frac{1}{d}\sum_{i=1}^{d}\lambda_i.$$

The product of the n eigenvalues is the determinant of the matrix $\mathbf{J}^T M\mathbf{J}$ and the sum of n eigenvalues is the trace of the matrix, so we obtain

$$\texttt{det}(\mathbf{J}^T M\mathbf{J})^{\frac{1}{d}} = \frac{1}{d} tr(\mathbf{J}^T M\mathbf{J}). \tag{2.11}$$

The equivalent form of this equation is $\mathbf{J}^{-T}\mathbf{J}^{-1} = \frac{1}{c}M$, where c is positive constant. This can be verified by taking the determinant and trace of $\mathbf{J}^T M\mathbf{J} = cI$, which implies $\mathtt{det}(\mathbf{J}^T M\mathbf{J})^{\frac{1}{d}} = c$ and $\frac{1}{d}tr(\mathbf{J}^T M\mathbf{J}) = c$. The eigenvectors of the matrix $J^{-T}J^{-1}$ specify the shape and orientation, or *alignment* of the mesh elements, as discussed above. This is controlled by M. In summary, for a given monitor function M satisfying the equidistribution (2.10) and alignment (2.11) conditions, we can control the size, shape, and orientation of the mesh elements through the physical domain.

Now, we will derive a functional for the inverse coordinate transformation $\xi = \xi(x) : \Omega \to \Omega_c$ that depends on the equidistribution (2.10) and alignment (2.11) conditions. Assume $\zeta_1, \zeta_2, ..., \zeta_d$ are the eigenvalues of the matrix $\mathbf{J}^{-1}M^{-1}\mathbf{J}^{-T}$. The well-known AM-GM inequality is used for the developing functional and gives us

$$\Big(\prod_{i=1}^{n} \zeta_i\Big)^{\frac{1}{d}} \leq \frac{1}{d}\sum_{i=1}^{d} \zeta_i. \tag{2.12}$$

The coordinate transformation that satisfies the approximate alignment condition minimizes this inequality. In other words, in order to satisfy the alignment condition we minimize the following difference

$$\frac{1}{d}\sum_{i=1}^{n} \zeta_i - \Big(\prod_{i=1}^{d} \zeta_i\Big)^{\frac{1}{d}}.$$

We have

$$\sum_{i=1}^{d} \zeta_i = \mathtt{tr}\Big(\mathbf{J}^{-1}M^{-1}\mathbf{J}^{-T}\Big) = \sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i,$$

and

$$\prod_{i=1}^{d} \zeta_i = \mathtt{det}\Big(\mathbf{J}^{-1}M^{-1}\mathbf{J}^{-T}\Big) = \frac{1}{(J\boldsymbol{\rho})^2}.$$

Equation (2.12) gives the following inequality

$$\Big(\frac{1}{J\boldsymbol{\rho}}\Big)^{\frac{2}{d}} \leq \frac{1}{d}\sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i$$

or equivalently,

$$d^{\frac{d\gamma}{2}}\frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^{\gamma}} \leq \boldsymbol{\rho}\Big(\sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i\Big)^{\frac{d\gamma}{2}}$$

for any real number $\gamma \geq 1$. Thus, the coordinate transformation determined by the alignment condition can be defined as a minimizer of the functional

$$\int_\Omega \left[\boldsymbol{\rho} \Big(\sum_i (\nabla \xi_i)^T M^{-1} \nabla \xi_i \Big)^{\frac{d\gamma}{2}} - d^{\frac{d\gamma}{2}} \frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^\gamma} \right] dx. \tag{2.13}$$

Moreover, in [14, 15] the inequality

$$\int_\Omega \frac{\boldsymbol{\rho}}{J\boldsymbol{\rho}} dx = \int_{\Omega_c} d\xi \leq \left(\int_\Omega \frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^\gamma} dx \right)^{\frac{1}{\gamma}},$$

is shown to be the special case of a more general inequality, for any real number $\gamma > 1$, with equality holding when $J\boldsymbol{\rho}$ is constant. Thus, the coordinate transformation based on equidistribution principle (2.10) can be defined as a minimizer of the following functional

$$\int_\Omega \frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^\gamma} dx - \left(\int_{\Omega_c} d\xi \right)^\gamma. \tag{2.14}$$

Neither functional (2.13) nor (2.14) alone provides a reasonable adaptive-mesh method and it is necessary to combine them together. For a given $\theta \in [0, 1]$, we can combine the functionals for the equidistribution and alignment conditions (2.13) and (2.14), as

$$\theta \int_\Omega \left[\boldsymbol{\rho} \Big(\sum_i (\nabla \xi_i)^T M^{-1} \nabla \xi_i \Big)^{\frac{d\gamma}{2}} - d^{\frac{d\gamma}{2}} \frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^\gamma} \right] dx + (1-\theta) d^{\frac{d\gamma}{2}} \left[\int_\Omega \frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^\gamma} dx - \left(\int_{\Omega_c} d\xi \right)^\gamma \right]. \tag{2.15}$$

Since $\int_{\Omega_c} d\xi$ is constant, we can omit this term from the functional. Hence, the functional which balances the equidistribution and alignment conditions is given by

$$I[\xi] = \theta \int_\Omega \boldsymbol{\rho} \cdot \Big(\sum_i (\nabla \xi_i)^T M^{-1} \nabla \xi_i \Big)^{\frac{d\gamma}{2}} dx + (1 - 2\theta) d^{\frac{d\gamma}{2}} \int_\Omega \frac{\boldsymbol{\rho}}{(J\boldsymbol{\rho})^\gamma} dx, \tag{2.16}$$

for $\theta \in [0, 1]$ and $\gamma > 1$.

Hence, an adaptive mesh is generated an image of a reference mesh using a coordinate transformation from the computational domain to the physical domain. The coordinate transformation can be determined by minimizing the functional (2.16).

To describe the equidistribution and alignment condition at the discrete level, we consider a mesh $\mathcal{T}_h$ of N elements with N_v vertices in the physical domain $\Omega \in \mathbf{R}^d$

$(d \geq 1)$. We consider an invertible affine mapping $F_K : \hat{K} \rightarrow K$ and its Jacobian matrix by $F_K^{'}$, where $\hat{K}$ is the reference or master element for a physical element K in $\mathcal{T}_h$. Assume that a metric tensor (or a monitor function) $\mathbb{M} = \mathbb{M}(x)$ is given on Ω which provides the shape, size, and orientation of mesh elements of the domain Ω. Generally, a mesh is uniform if all of its elements have the same size and is similar to a reference element $\hat{K}$. So, the main idea of the MMPDE method is to view any adaptive mesh $\mathcal{T}_h$ as a uniform mesh in the metric $\mathbb{M}$. The requirements of the equidistribution and alignment conditions can be expressed mathematically at the discrete level [17] as

$$|K|\sqrt{\mathtt{det}(\mathbb{M}_K)} = \frac{\sigma_h}{N}, \forall K \in \mathcal{T}_h,$$

$$\frac{1}{n}tr((F_K^{'})^T\mathbb{M}_K F_K^{'}) = \mathtt{det}((F_K^{'})^T\mathbb{M}_K F_K^{'})^{\frac{1}{d}}, \forall k \in \mathcal{T}_h,$$

where $|K|$ is the volume of K and $\sigma = \sum_{K\in\mathcal{T}_h} |K|\sqrt{\mathtt{det}(\mathbb{M}_k)}$.

A standard choice of $\mathbb{M}$ is the symmetric and positive definite piecewise function

$$\mathbb{M}_K = \det(I + \alpha_h^{-1}|H_K|)^{-\frac{1}{d+4}}(I + \alpha_h^{-1}|H_K|), \forall K \in \mathcal{T}_h, \tag{2.17}$$

where $\mathcal{T}_h$ is the physical mesh and H_K is the approximate Hessian of the solution on element K. H_K can be calculated by a least-squares Hessian recovery technique, and the regularity parameter α_h is chosen so that

$$\sum_{K\in\mathcal{T}_h} |K|\sqrt{\det(\mathbb{M}_K)} = 2|\Omega|.$$

The choice of $\mathbb{M}$ in (2.17) is known to be optimal with respect to the L_2 norm of the linear interpolation error [16]. The mesh points will concentrate where the determinant of the Hessian is large.

A discrete functional associated with the equidistribution and alignment conditions is given by

$$I[\mathcal{T}_h] = \sum_{K\in\mathcal{T}_h} |K|\sqrt{\det(\mathbb{M}_K)}\Big[\theta\Big(\mathrm{tr}(\mathbb{J}\mathbb{M}_K^{-1}\mathbb{J}^T)\Big)^{\frac{d\gamma}{2}} + (1-2\theta)d^{\frac{d\gamma}{2}}\Big(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}_K)}}\Big)^{\gamma}\Big], \tag{2.18}$$

where $\mathbb{J} = (F_K^{'})^{-1}$. Minimizing this functional $I[\mathcal{T}_h]$ approximately satisfies the equidistribution and alignment conditions [15]. The value of the parameters $\theta = \frac{1}{3}$, and $\gamma = \frac{3}{2}$ are used for our numerical experiments.

Moving Mesh PDE The moving mesh PDE (MMPDE) equation can be defined as the (modified) gradient system (or gradient flow equation) for the energy functional, i.e.,

$$\frac{d\mathbf{x}_i}{dt} = -\frac{P_i}{\tau}\frac{\partial I[\mathcal{T}_h]}{\partial \mathbf{x}_i}, \quad i = 1, 2, .., N_v, \quad t \in (t_n, t_{n+1}], \tag{2.19}$$

where $P_i = \det(\mathbb{M}_i)^{\frac{1}{d+2}}$ is a scalar function that can be used to make the equation have invariance properties and τ is a positive parameter used to adjust the response time of mesh movement to the change in $\mathbb{M}$. A smaller value of τ provides faster response. $\tau = 10^{-4}$ is used for our computation.

2.1.3 Choosing the mesh density function

The Hessian based monitor function is a great choice for some problems by setting proper values of the parameters. Sometimes we require modification of the Hessian - based metric tensor in order to achieve a level of mesh concentration around a specific geometric location and the moving front of the domain. A reasonable way to modify the standard Hessian-based metric tensor is by designing a function β based on a specific location or the moving front and then combining it with the Hessian-based metric tensor as follows

$$\tilde{\mathbb{M}}_K \simeq \mu\mathbb{M}_K + \beta I, \tag{2.20}$$

where μ is a positive scalar used to adjust the contribution of the Hessian based metric tensor (2.17) and β is designed based on the specific location or moving front. Here are some examples and techniques for designing β, which aim to achieve a reasonable mesh concentration around the moving boundary or a specific geometric region Γ_q.

Example 2.1.1. Consider a rectangular domain Ω containing a circular hole Γ_q centered at (h, k) with radius r. To achieve a reasonable mesh concentration around the Γ_q, β is designed as follows

$$\beta = \left[e^{4(r-\sqrt{(x-h)^2+(y-k)^2})} - 1 + \frac{1}{\max_{K\in\mathcal{T}_h}\sqrt{\det(\mathbb{M}_K)}} \right]^{-1},$$

and $\mu = 1$ is chosen for (2.20). The β function gives a larger positive value on a circle.

We can easily observe that for mesh points (x, y) on the hole or circle gives us

$$\tilde{\mathbb{M}}_K \simeq \mathbb{M}_K + \max_{K \in \mathcal{T}_h} \sqrt{\det(\mathbb{M}_K)}.$$

This will provide a mesh concentration around the circle which is similar to the concentration in regions with the largest value of $\sqrt{\det(\mathbb{M}_K)}$. The exponential term in β ensures that the concentration decreases rapidly enough to prevent over concentration of mesh elements near the circle [9].

Example 2.1.2. Consider a rectangular domain Ω with a circular void in the domain centered at (h, k) and the radius of the void r. To achieve a reasonable mesh concentration around the void and a relatively uniform mesh inside the void, we can design β as follows

$$\beta = \begin{cases} \nu, & \text{if} \quad ((x-h)^2 + (y-k)^2) \leq r^2 \\ \left[e^{\nu(r - \sqrt{(x-h)^2 + (y-k)^2})} - 1 + \frac{1}{\max_{K \in \mathcal{T}_h} \sqrt{\det(\mathbb{M}_K)}} \right]^{-1}, & \text{otherwise,} \end{cases}$$

where ν is a positive constant. This β is a piecewise defined function combining both locations inside and outside of the circular void. The constant ν will ensure a relatively uniform mesh distribution inside the void and the other function in β will provide a high mesh concentration near the edges of the circular void.

Example 2.1.3. Consider a rectangular type domain $\Omega = [-1, 1] \times [0, 1] \bigcup \Gamma_q$ which has a semicircular boundary that is situated at the bottom boundary of the rectangle, where the semi-circle is centered at $(h = 0, k = 0)$ with radius $r = 0.2$. To achieve the reasonable mesh concentration around the semi-circle, β is designed as follows

$$\beta = \left[\min_{\Gamma_q} \Big(||(x, y) - (x_q, y_q)||_2 \Big)^2 + \frac{1}{\sqrt{N}} \right]^{-1}, \quad (x_q, y_q) \in \Gamma_q,$$

where N is the total number of mesh points. This is known as a distance-based function, where β is maximum on the boundary of the semi-circle. A combination of the standard Hessian-based approach and a distance-based approach is able to provide a reasonable mesh concentration near the semi-circle.

2.1.4 Implementation of mesh and physical solve

There are two approaches that can be used to solve the physical PDE and the mesh equation: simultaneously or alternately. In a simultaneous solution approach, the discrete physical PDE and the mesh equation provide a coupled system. Figure 2.2 represents the simultaneous solution strategy. The main disadvantage of this approach is that it has highly nonlinear coupling between the physical solution and the mesh. On the other hand, an alternating solution approach generates the mesh $\mathbf{x}^{n+1}$ at a new time step using the physical solution φ^n and the mesh $\mathbf{x}^n$ at the current time, and then computes the solution φ^{n+1} at the new time step. This is shown in Figure 2.3. In this approach, there is a lags in time because $\mathbf{x}^{n+1}$ depends on the solution and the mesh at the previous time. This does not create any trouble if the time step is reasonably small. The main advantages of the alternating approach are the mesh generation code is used separately so it is flexible and improves the efficiency at each step, and the mesh adaptation is not tightly coupled with the physical PDE. The simultaneous solution approach is mainly used to solve one-dimensional problems, and the alternating solution approach is mainly used to solve multidimensional problems.

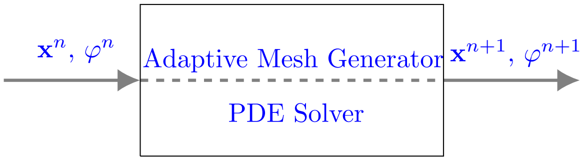

$\mathbf{x}^n$, φ^n
Adaptive Mesh Generator
PDE Solver
$\mathbf{x}^{n+1}$, φ^{n+1}

Figure 2.2: Simultaneous solution approach.

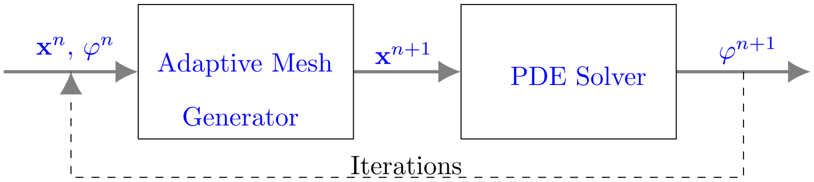


Figure 2.3: Alternate solution approach.

The alternating solution procedure is used for our model problem.

2.2 Finite element method

The Finite Element Method (FEM) is a widely used numerical method to approximate the solution of PDEs. If the PDE is time dependent, then we can convert the problem to a system of ordinary differential equations once the PDE is discretized spatially by the FEM method. This system of ordinary differential equation can be integrated numerically by well known techniques. The main advantage of this approach is separate treatment of the spatial and temporal components of the PDE. This technique is also known as the *method of lines*.

To illustrate the finite element method, we consider the well-known one-dimensional Burgers' equation, which is a nonlinear PDE that describes the motion of a viscous fluid,

$$u_t = \varepsilon u_{xx} - \left(\frac{u^2}{2}\right)_x, \quad x \in (0,1), \quad t > 0, \tag{2.21}$$

subject to the following boundary conditions

$$u(0,t) = u(1,t) = 0, \tag{2.22}$$

and initial condition

$$u(x,0) = \frac{0.1e^{\frac{-x+0.5}{20\varepsilon}} + 0.5e^{\frac{-x+0.5}{4\varepsilon}} + e^{\frac{-x+0.375}{2\varepsilon}}}{e^{\frac{-x+0.5}{20\varepsilon}} + e^{\frac{-x+0.5}{4\varepsilon}} + e^{\frac{-x+0.375}{2\varepsilon}}}, \tag{2.23}$$

where ε is a positive physical parameter. For our experimental purposes, ε is chosen as 10^{-4}.

2.2.1 Finite element method on a fixed mesh

To obtain a numerical solution using the FEM based on the Galerkin formulation for differential equations, the following procedure is used. The procedure begins by deriving a variational or weak formulation by multiplying both sides of the PDE (2.21) by a test function $\phi(x,t)$, which satisfies the boundary conditions $\phi(0,t) = \phi(1,t) = 0$

and then integrating in space from $x = 0$ to $x = 1$. Then, we find $u(\cdot, t) \in V$ for $t > 0$ such that

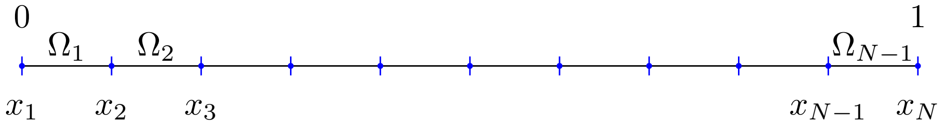
$$\begin{aligned}\int_0^1 u_t \phi dx &= \int_0^1 \Big(\varepsilon u_x - \Big(\frac{u^2}{2}\Big)_{xx}\Big)\phi dx, \\ &= \varepsilon \phi u_x\Big|_0^1 - \int_0^1 \varepsilon \phi_x u_x dx - \phi\frac{u^2}{2}\Big|_0^1 + \int_0^1 \phi_x \frac{u^2}{2} dx, \\ &= \int_0^1 \Big(-\varepsilon u_x + \frac{1}{2}u^2\Big)\phi_x dx, \quad \forall \phi \in V, 0 < t \leq T \end{aligned} \tag{2.24}$$

where V is the solution space, and is defined by

$$V = H_0^1(0,1) \equiv \{u | u \in H^1(0,1), u(0) = u(1) = 0\}.$$

Equation (2.24) is the weak or variational form of the PDE. The problem is still in continuous form in an infinite dimensional space V. In order to solve the problem numerically, we need to represent the problem in a finite dimensional space.

In the second step, we generate a finite element mesh. We divide the domain $\Omega = [0, 1]$ into finite elements or non-overlapping subintervals $\Omega_e, e = 1, 2, ..., N - 1$



such that

$$\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_{N-1}.$$

We introduce a set of points on each element called nodes, which are uniformly spaced throughout the element. For a given positive integer N, we can define the mesh

$$\mathcal{T}_h = \{x_j | x_j = (j-1)h, \quad j = 1, ..., N\}$$

where $h = \frac{1}{N-1}$. Thus, the nodes x_j and the elements $\Omega_e, e = 1, 2, ..., N - 1$ uniquely define a finite element mesh on which we obtain our discrete representation of the solution.

In the third step, we construct a finite dimensional space V^h, which is a subspace of V, i.e., $V_h \subset V$. Since V_h has finite dimension, we choose a set of basis functions that are linearly independent. Thus, V^h is spanned by the basis functions. We wish to use standard piecewise linear basis functions for our approximations on the uniform mesh $\mathcal{T}_h$ are represented by

$$\phi_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}, & \text{for } x \in [x_{j-1}, x_j], \\ \frac{x_{j+1}-x}{x_{j+1}-x_j}, & \text{for } x \in [x_j, x_{j+1}], \qquad j = 1, \ldots, N \\ 0, & \text{otherwise.} \end{cases}$$

For our problem, we assume V^h is the $(N-2)$ dimensional subspace of V, which is spanned by the linear basis functions $\phi_2, \ldots, \phi_{N-1}$, such that

$$V^h = \operatorname{span}\{\phi_2, \ldots, \phi_{N-1}\}.$$

In the fourth step, we represent a finite element solution using a linear combination of the basis functions. A finite element approximation $u^h(\cdot, t) \in V^h$ for $0 < t \leq T$ of the physical problem must satisfy

$$\int_0^1 u_t^h \phi dx = \int_0^1 \Big(-\varepsilon u_x^h + \frac{1}{2}\left(u^h\right)^2 \Big) \phi_x dx, \quad \forall \phi \in V^h, 0 < t \leq T, \tag{2.25}$$

and the initial conditions

$$u^h(x_j, 0) = u(x_j, 0), \quad j = 1, \ldots, N. \tag{2.26}$$

Now, we form an approximation of $u(x,t)$ in the finite dimensional space V^h by setting u^h to be a linear combination of the basis functions, i.e.:

$$u^h(x,t) = \sum_{j=2}^{N-1} u_j(t)\phi_j(x), \tag{2.27}$$

where $u_j(t)$ represents the nodal unknown value of $u(x_j, t)$ at the jth node at time t, i.e., $u_j(t) \approx u(x_j, t)$. Now, taking $\phi = \phi_k(x), k = 2, \ldots, N-1$ gives us

$$\sum_{j=1}^{N} \frac{du_j}{dt} \int_0^1 \phi_j(x)\phi_k(x) dx = -\varepsilon \sum_{j=1}^{N} u_j \int_0^1 \phi_j'(x)\phi_k'(x) dx + \frac{1}{2}\int_0^1 \Big(\sum_{j=1}^{N} u_j \phi_j(x) \Big)^2 \phi_k'(x) dx,$$

$$k = 2, \ldots, N-1. \tag{2.28}$$

In order to implement the ordinary differential equation solver, the boundary conditions

$$u_1^h(t) = 0, \qquad u_N^h(t) = 0, \qquad t > 0$$

can be put in the ordinary differential equation form

$$\frac{du_1^h}{dt} = 0, \qquad \frac{du_N^h}{dt} = 0, \qquad t > 0. \tag{2.29}$$

The set of equations presented in (2.28) and boundary conditions (2.29) form a nonlinear system of N ordinary differential equations.

In the final step of our implementation of the FEM, we solve the system of ordinary differential equations. The initial value problem consisting of the ordinary differential equation system (2.28) and (2.29), with initial conditions (2.26), can be solved using a standard ordinary differential equation solver, which gives the solution for the nodal unknown variables $u_1(t), ..., u_N(t)$.

2.2.2 Finite element method on an adaptive moving mesh

To achieve an adaptive solution for the model problem we wish to place mesh points in the steep front of the solution and continually adjust the mesh as the front propagates in time. Such a mesh is called an *adaptive moving mesh.* To better understand adaptive mesh movement, we can implement a coordinate transformation. In particular, consider that we have a time-dependent coordinate transformation, denoted as $x = x(\xi, t) : \Omega_c \equiv [0, 1] \longrightarrow \Omega \equiv [0, 1]$, where Ω_c and Ω represent the computational and physical domains, respectively. This transformation is chosen in such a way that the solution can be represented using the transformed spatial variable,

$$\hat{u}(\xi, t) = u(x(\xi, t), t).$$

A corresponding moving mesh can be presented as

$$\mathcal{T}_h(t) : \quad x_j(t) = x(\xi_j, t), \quad j = 1, ..., N$$

for the fixed, uniform mesh on Ω_c,

$$\mathcal{T}_h^c : \quad \xi_j = \frac{j-1}{N-1}, \quad j = 1, ..., N.$$

The basis functions and the approximation function space are now time-dependent

$$\phi_j(x,t) = \begin{cases} \frac{x-x_{j-1}(t)}{x_j(t)-x_{j-1}(t)}, & \text{for } x \in [x_{j-1}(t), x_j(t)]\,, \\ \frac{x_{j+1}(t)-x}{x_{j+1}(t)-x_j(t)}, & \text{for } x \in [x_j(t), x_{j+1}(t)]\,, \quad j = 1, \ldots, N, \\ 0, & \text{otherwise,} \end{cases} \tag{2.30}$$

and the finite dimensional space is also time dependent:

$$V^h(t) = \text{span}\left\{\phi_2(\cdot,t), \ldots, \phi_{N-1}(\cdot,t)\right\}. \tag{2.31}$$

We now find $u^h(\cdot,t) \in V^h(t)$ for $0 < t \leq T$ that satisfies

$$\int_0^1 u_t^h \phi dx = \int_0^1 \left(-\varepsilon u_x^h + \frac{1}{2}\left(u^h\right)^2\right) \phi_x dx \quad \forall \phi \in V^h(t), 0 < t \leq T, \tag{2.32}$$

and the initial condition

$$u^h(x_j(0), 0) = u(x_j(0), 0), \quad j = 1, ..., N. \tag{2.33}$$

Now, the FEM approximation, u^h, is time dependent so we need a special treatment for the time derivative of u^h. The form of the approximation is given by

$$u^h(x,t) = \sum_{j=1}^{N} u_j(t)\phi_j(x,t), \tag{2.34}$$

where $u_j(t) \approx u(x_j(t), t)$. Now, differentiating both sides with respect to t, we get the time derivative using the chain rule as

$$u_t^h(x,t) = \sum_{j=1}^{N} \left(\frac{du_j}{dt}(t)\phi_j(x,t) + u_j(t)\frac{\partial \phi_j}{\partial t}(x,t)\right). \tag{2.35}$$

In similar fashion, the time derivative of the basis function ϕ_j becomes

$$\frac{\partial \phi_j}{\partial t}(x,t) = -\frac{\partial \phi_j}{\partial x}(x,t) X_t(x,t), \tag{2.36}$$

where $X_t(x;t)$ is the linear interpolant of the nodal mesh speeds, i.e.,

$$X_t(x,t) = \sum_{j=1}^{N} \frac{dx_j}{dt}(t)\phi_j(x,t). \tag{2.37}$$

Therefore, we have

$$\begin{aligned} u_i^h(x,t) &= \sum_{j=1}^{N}\left(\frac{du_j}{dt}(t)\phi_j(x,t) - u_j(t)\frac{\partial \phi_j}{\partial x}(x,t)X_t(x,t)\right) \\ &= \sum_{j=1}^{N}\frac{du_j}{dt}(t)\phi_j(x,t) - \frac{\partial u^h}{\partial x}(x,t)X_t(x,t). \end{aligned} \tag{2.38}$$

Inserting this expression into (2.32) we obtain

$$\int_0^1 \left(\sum_{j=1}^{N}\frac{du_j}{dt}\phi_j - \frac{\partial u^h}{\partial x}X_t\right)\phi dx = \int_0^1 \left(-\epsilon u_x^h + \frac{1}{2}\left(u^h\right)^2\right)\phi_x dx. \tag{2.39}$$

We obtain a system of ODEs for $u_1(t), u_2(t), \ldots, u_N(t)$ by substituting $\phi = \phi_k(x,t)$ into (2.39) for $k = 2, ..., N-1$,

$$\int_0^1 \left(\sum_{j=1}^{N}\frac{du_j}{dt}\phi_j - \frac{\partial u^h}{\partial x}X_i\right)\phi_k dx = \int_0^1 \left(-\varepsilon u_x^h + \frac{1}{2}\left(u^h\right)^2\right)(\phi_k)_x dx, \tag{2.40}$$

with the boundary conditions given by equation (2.29) and the initial condition (2.23), the system can be integrated in time.

The resulting system is solved along with the mesh equations (2.5) incorporating the boundary conditions and a suitable choice of monitor function for the adaptive numerical solution.

2.2.3 An example: Burgers' equation with an exact solution

To illustrate mesh adaptation a using moving mesh, we consider the well-known one-dimensional Burgers' equation (2.21) together with boundary conditions (2.22) and initial condition (2.23). The initial condition has two steep fronts, which are known as shock waves in a fluid, and the waves are traveling towards the right. As time progresses, the two shock fronts gradually move closer together, finally merging and forming a steeper shock wave around t = 0.55. This phenomenon of two waves merging is commonly observed in fluid mechanics and can result in the formation of complex wave patterns. The merging of these two fronts may substantially impact the

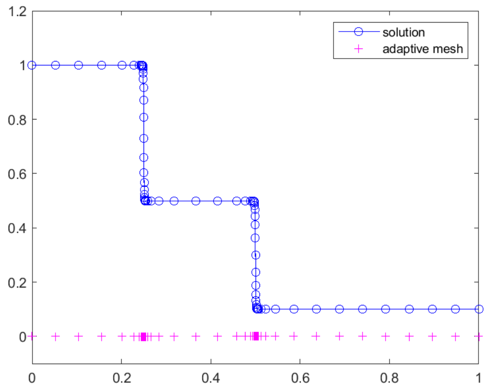
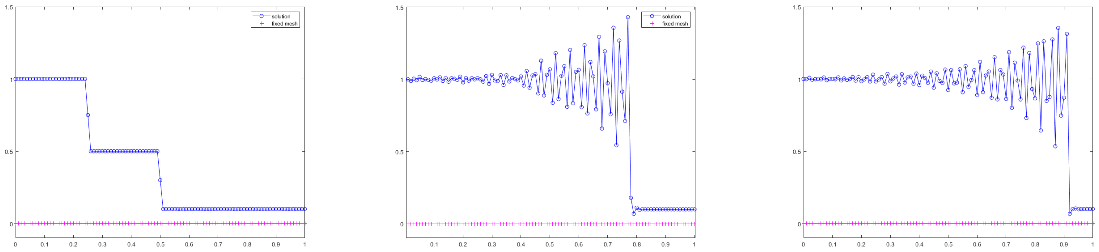


Figure 2.4: A plot of the initial solution (2.41) displayed on an adaptive initial mesh comprising 71 points.

behavior of the fluid system and may cause an increase or decrease in the steepness of resulting shock wave.

The exact solution of the Burgers' equation (2.21) with boundary and initial conditions is

$$u(x,t) = \frac{0.1e^{\frac{-x+0.5-4.95t}{20\varepsilon}} + 0.5e^{\frac{-x+0.5-0.75t}{4\varepsilon}} + e^{\frac{-x+0.375}{2\varepsilon}}}{e^{\frac{-x+0.5-4.95t}{20\varepsilon}} + e^{\frac{-x+0.5-0.75t}{4\varepsilon}} + e^{\frac{-x+0.375}{2\varepsilon}}}, \tag{2.41}$$



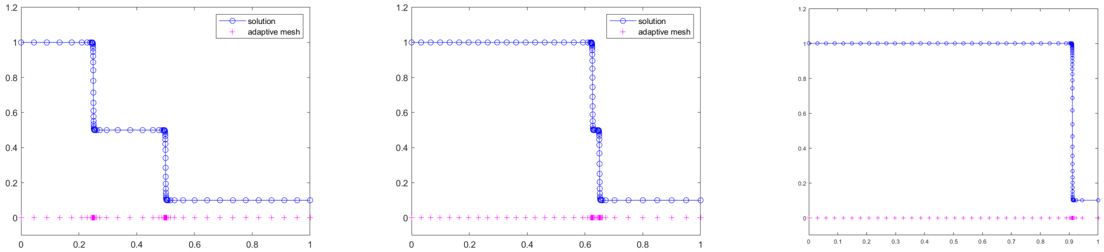
(a) $t = 0$ s

(b) $t = 0.5$ s

(c) $t = 1$ s

Figure 2.5: FEM solution on a fixed mesh consisting of 101 points.

Figure 2.5 represents the numerical results on a fixed mesh having 101 points and it is observed that the numerical solution still contains oscillations near the steep fronts. Figure 2.6 demonstrates the moving mesh approach with 81 mesh points,



(a) $t = 0$ s

(b) $t = 0.5$ s

(c) $t = 1$ s

Figure 2.6: FEM solution on a adaptive moving mesh consisting of 81 points.

which prevented the oscillations near steep fronts. Note that the Hessian-based function $\boldsymbol{\rho} = \left(1 + \frac{1}{\alpha}|u_{xx}|^2\right)^{1/3}$ is used as a monitor function in the mesh equation.

2.3 Corrosion

2.3.1 Corrosion Basics

Corrosion is the degradation of a metal surface caused by chemical or electrochemical reactions with its surrounding environment. Metals mostly corrode by electrochemical reactions and non-metals corrode by chemical reactions. A chemical reaction is a process where the reactants are converted into different products. However, at least one of the species changes its number of valance electrons during the electrochemical reaction. As our interest is in metallic corrosion, the following discussion will primarily focus on this topic.

2.3.1.1 Corrosion electrochemistry

Usually, metals are in high energy state and have a tendency to go to a lower energy state by forming oxides or hydroxides depending on their environment. The tendency to oxidize depends on the position of the metal in the Galvanic series table [20, 22], with metals located at the top of the galvanic series being more active and typically acting as an anode. It is the anode that is usually corroded in its environment.

Corrosion processes in metals occur due to electrochemical reactions in a corrosion cell. Within the corrosion cell, there are mainly two reactions that occur and the generic forms of the reactions are

$$\begin{aligned} \mathrm{M} &\longrightarrow \mathrm{M^{n+}} + n\mathrm{e^-} \text{ (anodic reaction or oxidation)} \\ n\mathrm{X^+} + n\mathrm{e^-} &\longrightarrow \mathrm{X_n} \text{ (cathodic reaction or reduction)} \end{aligned}$$

where M is the generic metal atom, $\mathrm{M^{n+}}$ is the dissolved metal ion with a charge number of n, and $\mathrm{e^-}$ is an electron. There are four necessary components for any metallic corrosion to occur [22]:

1. There must be a sacrificial metal that tends to corrode called the anode.
2. There must be a less corrosive conductive metal called the cathode.
3. The anode and cathode should be in direct physical contact (or by a wire) to allow electrons to flow from anode to cathode.
4. The anode and cathode must be exposed to an electrolyte that provides a conductive liquid path, which allows ions to exchange between anode to cathode and hence completing an electric circuit.

These four elements: anode, cathode, electrical path, and electrolyte, form a corrosion cell. The corrosion process cannot occur if any corrosion cell elements are absent. Figure 2.7 presents an example of a corrosion cell for iron.

The most common electrolytes are water, acid, and alkaline. Tap water is less corrosive than rainwater because rainwater has a lower pH level, which indicates a higher acid content. Some water is even more corrosive than rainwater due to its high salinity, which makes a stronger electrolyte. Many conditions influence the initiation and impact of corrosion, such as the type of electrolyte, the availability of oxygen, the time of exposure, the temperature, the stress, and other biological conditions. One of the most corrosive environments is found in coastal regions with high humidity and salt concentration, creating an especially strong electrolyte.

The process of corrosion involves: the material dissolving by electrochemical reactions on the metal surface in the presence of an electrolyte, and then the dissolved

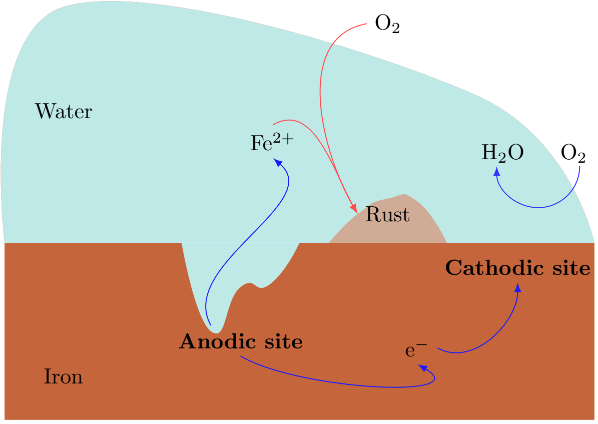


Figure 2.7: A basic depiction of the corrosion cell and the process of pitting corrosion of iron. Figure adapted from [5].

metal atoms traveling into the electrolyte or the surrounding metal surface. During the electrochemical reactions, the anodic reaction produces electron(s); the electrons travel in the metal and reach (generally outside of the corroded area) a point on the surface and then participate in a cathodic reaction. The resulting cathodic reactions produce a corrosion product. The impact of corrosion is the loss of mechanical strength, which leads to physical damage or degradation of metals [18].

In the corrosion process, there is a transfer of electrons and ions between two metal areas, and these transfers can happen in two ways. The first way is when two connected metals, with sufficient potential differences, are in an electrolyte. The metal with a higher potential becomes an anode which may corrode faster, and the metal with a lower potential becomes a cathode that may corrode slower. The second way is when different areas on the same metal have enough potential difference, allowing ions and electrons to exchange. This phenomenon is commonly known as *self-corrosion* and can occur for many reasons, such as from differences in microstructure, composition, or impurities in the metal [18]. An example of self-corrosion is presented in Figure 2.7. It can take a continuum of forms, from general corrosion to localized corrosion.

General corrosion leads to the metal surface corroding uniformly over the entire surface area, resulting in reduced mass and even thinning of the metal. A familiar example of general corrosion is rusting of an iron fence. In the case of localized corrosion, corrosion only occurs in specific area(s) and most of the areas of the metal surface may remain unaffected. Localized corrosion can occur in different forms such as pitting corrosion, crevice corrosion, and stress corrosion cracking. These forms of corrosion can be more dangerous than general corrosion because they can cause localized damage and lead to failures in engineering systems.

Metals can experience localized corrosion due to the breakdown of a protective layer, also known as the passivation layer, such as is present in stainless steel. This protective layer is a barrier between the metal surface and the electrolyte, preventing electrochemical reactions from occurring on the metal surface. Regardless, when the protective layer is damaged or broken, the metal surface is exposed to the electrolyte and forms a corrosion cell. Electrochemical reactions occur in the cell, and the anode is highly localized in this situation. As a result, the anode undergoes corrosion and penetrates the depth of the metal, forming a tiny pit, hole, or cavity, whereas the surrounding metal surface remains unaffected. This form of corrosion is called *pitting corrosion* [18, 27].

2.3.1.2 The electric potential, corrosion potential and applied potential

The electric potential, also known as the electrostatic potential, represents the amount of energy needed to move or transport a unit of electric charge within an electric field from a reference point to a specific point. Note that the electric potential is obtained at a single point, whereas the potential difference is measured between two distinct points.

The corrosion potential is a measure of a metal's tendency to corrode; in other words, the corrosion potential is the tendency of the metal to lose electrons (oxidation) in the presence of the electrolyte. However, it does not provide a corrosion rate directly; the corrosion rate is proportional to the electron transfer between electrodes in the electrolyte. Two electrodes are formed spontaneously during the corrosion process, an anode and a cathode [34]. Normally, electrons move from the electrode with the most negative potential to the one with the most positive potential in a circuit, and the current flows in the opposite direction. Electric potential for general metals are found using the known potential for "standard" electrodes. The standard electrode in a laboratory is the Standard Hydrogen Electrode (SHE) and Table 2.1 gives the electrode potentials with respect to the SHE. It shows that zero potential is at the hydrogen electrode. The electrode potential determines the direction of electron flow. Therefore, given the electrode potentials of two metals in contact, one can evaluate the anode and cathode of the electrochemical cell. For instance, from the table, copper has a higher potential than iron and electrons will transfer from iron to copper. According to the above discussion of corrosion phenomena, iron will behave as the anode and undergo corrosion [18].

Table 2.1: Electrode potential relative to SHE [19]

Reaction	V
$Cu^{2+} + 2e^{-} \rightarrow Cu$	+0.342
$2H^{+} + 2e^{-} \rightarrow H_2$	0.00
$Fe^{2+} + 2e^{-} \rightarrow Fe$	-0.447

Applied potential is also known as an external electrical potential, which is applied intentionally to a system. It is an external force that influences the movement of electrons within the system. One can control the electrochemical reactions at the electrode-electrolyte interface by applying a specific potential. This applied potential can alter the rate and direction of electron transfer, consequently affecting corrosion, electrolysis, and electroplating processes. The chemist uses the applied potential for different tests in the electrochemical industry. One example is accelerating the electrochemical corrosion testing processes. In summary, the applied potential deliberately manipulates the electrical potential to regulate or induce desired reactions in an electrochemical system.

2.3.2 Pitting corrosion

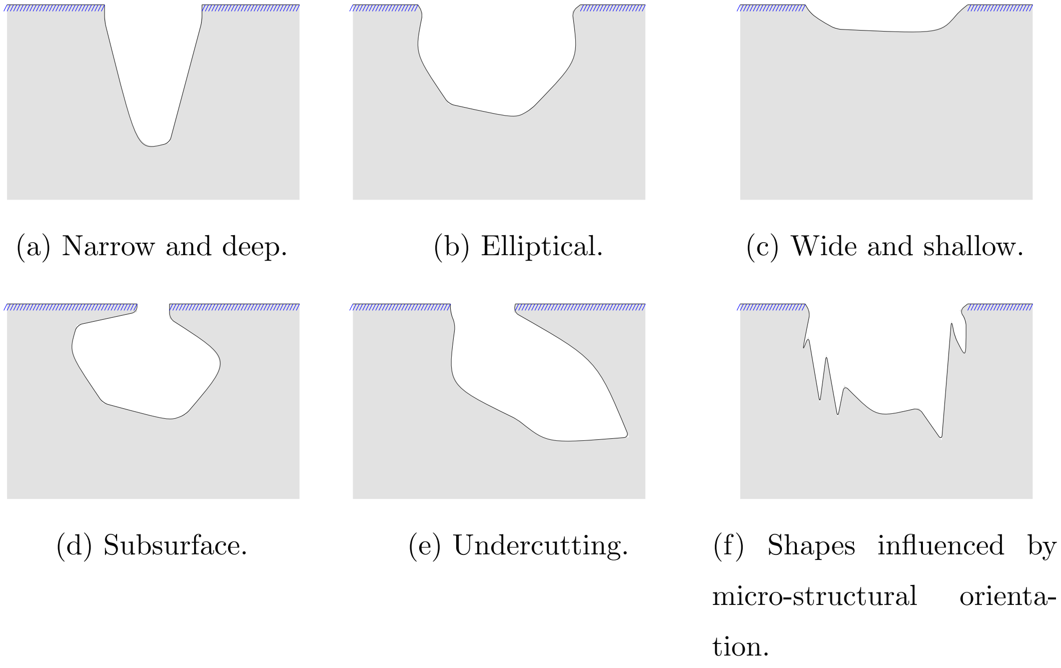
2.3.2.1 Concept of pitting corrosion

Pitting corrosion is a localized form of corrosion that generates pits, cavities, or holes in the metal. It occurs when the protective or passive layer of the metal surface breaks down. Usually, the passive layer prevents the metal surface from undergoing chemical reactions with its surrounding environment and the metal does not corrode. When the layer is broken, the metal surface is exposed to the environment forming a corrosion cell (in the presence of electrolyte) in that specific area and initiating the corrosion process. The corrosion process is highly localized and penetrates the metal. As a result, small pits, cavities, or holes are formed on the metal surface, whereas the other parts of the surface remain protected. This phenomenon is known as *pitting corrosion* and the mechanism of pitting corrosion is demonstrated in Figure 2.7.

Hence, pitting corrosion occurs through electrochemical corrosion, as discussed in the previous section. When a small part of the passive layer breaks, the metal surface is exposed to the environment and electrochemical reactions take place if the environment is conductive to the exchange of current. Electrons move from the exposed area (which is the active or anodic area) to the surrounding area or cathodic area, which is highly passive. If the cathodic area is very large, the demand for electrons is high and cannot be fulfilled by the small anodic area. Thus, the localized attack is caused by the large difference between the passive surface area and the active (or anodic) area inside the pit. This causes corrosion in the depth direction. The danger of pitting corrosion lies in the defects that form on the metallic body during the formation of the pits. These pits are regions of high stress and reduce the strength of the metal, shortening its service life. Additionally, leaks can be caused by pitting that penetrates too deeply into the metal leading to further physical damage and degradation of the metal [18, 27].

2.3.2.2 Pit shapes and growth

Pitting corrosion generates pits on the metal surface that can have various shapes, including hemispherical, cup-shaped, wide-and-shallow, or deep and narrow troughs, etc. [13, 22]. The shape and growth rate of the pit is influenced by the type of metal and its environment. Figure 2.8 presents examples of commonly observed pitting shapes in experiments.


(a) Narrow and deep.

(b) Elliptical.

(c) Wide and shallow.

(d) Subsurface.

(e) Undercutting.

(f) Shapes influenced by micro-structural orientation.

Figure 2.8: Some examples of common shapes of pitting corrosion. Figure adapted from [13].

Several studies have indicated three distinct stages in the development of pitting corrosion: the initiation stage, the metastable stage, and the stable stage. The initiation of a pit is followed by metastable growth that leads to stable growth under the right balance between electrochemical and mass transport mechanisms. The initiation stage of pitting corrosion is not well understood and continues to be a topic of significant discussion and research [13, 22, 24]. After the passive layer breaks down and a pit initiates, the pit may or may not be stable, which is referred to as a metastable pit. A pit is identified as stable if it is actively growing over time [1, 8]. In this thesis work, we consider the pit to be in the stable stage.

2.3.2.3 Transport in solution

In the corrosion process, species move or transport, in the solution or are deposited nearby. Under certain circumstances, the rate of corrosion is controlled by the speed of the transport process, which depends on diffusion, migration, and convection within the electrolyte. Diffusion and migration are the two process involved primarily in pitting corrosion [27].

Diffusion is a movement of species driven by concentration gradients. Usually, species move from a higher concentration to a lower concentration area until all species reach an even concentration or equilibrium state. Fick's first law states that the diffusion flux $N_{\text{diffusion}}$ is directly proportional to the concentration gradient [11, 12],

$$\begin{aligned}\mathbf{N}_{j,\text{diffusion}} &\propto \nabla c_j \\ &= -D_j \nabla c_j\end{aligned}$$

where ∇ is the gradient operator (i.e, $\nabla = \hat{i}\frac{\partial}{\partial x} + \hat{j}\frac{\partial}{\partial y}$ in two dimensions), D_j is diffusion coefficient or diffusivity of species j, and c_j is the concentration of species j. The negative sign indicates that the flow occurs from a higher to a lower concentration region. In our work, the flux will be assumed to be independent of time.

Migration is a movement of a charged species driven by an electric field. In other words, migration is the phenomenon in which charged species are transported across a solution due to the presence of an electric voltage gradient, commonly referred to as the potential gradient. The flux of a charged species j induced by migration in the solution is expressed by [2, 27]:

$$\mathbf{N}_{j,\text{migration}} = -z_j \frac{F D_j}{RT} c_j \nabla \varphi$$

where z_j is the charge number of the species j, φ is the electric potential, F is Faraday's constant (9.649×10^4 C/mol), R is the universal gas constant (8.314 $\text{Jmol}^{-1}\text{K}^{-1}$) and T is the absolute temperature (294K).

Convection occurs when a fluid or solution moves from one location to another under the influence of applied forces.

$$\mathbf{N}_{j,\text{convection}} = c_j\mathbf{v}$$

The Nernst-Plank equation (2.42) is a widely used equation for describing the transport flux of chemically charged species [3] within an electrolyte:

$$\mathbf{N}_j = -D_j\nabla c_j - z_j u_j F c_j \nabla\varphi + c_j\mathbf{v} \tag{2.42}$$

where $\mathbf{N}_j$ is the flux of species j, c_j is the concentration of species j, $\mathbf{v}$ is the solvent velocity, and u_j is the mobility of the species j, which follows the expression

$$u_j = \frac{D_j}{RT}.$$

Hence, the right side of (2.42) represents three fluxes; the first term -$D_j\nabla c_j$ expresses the transport rate of ionic diffusion flux, the second term -$z_j u_i F c_j \nabla\varphi$ describes the electro-migration flux for charge species, and the third term $c_j\mathbf{v}$ represents the convection flux.

Computational governing equation: The transport of species in the solution can be derived from the conservation of mass principle [3, 21, 28], which is represented by the equation

$$\underbrace{\frac{\partial c_j}{\partial t}}_{\text{Storage}} = \underbrace{-\nabla\cdot\mathbf{N}_j}_{\text{Flux in- Flux out}} + \underbrace{R_j}_{\text{Generation}}$$

where R_j is the rate of production of species generation due to chemical reactions. We used a negative sign in front of the $\nabla\mathbf{N}_j$ term because we are considering flux in minus flux out. Substituting $\mathbf{N}_j$ into the previous equation gives the following equation

$$\frac{\partial c_j}{\partial t} = -D_j\nabla^2 c_j - z_j u_i F\nabla(c_j\nabla\varphi) + \nabla(c_j\mathbf{v}) + R_j \tag{2.43}$$

This equation represents the rate of change of concentration driven by the diffusion, migration, convection, and rate of production of species.

The Nernst-Planck equation is solved for each species involved in the model and when electromigration is included in the model, an extra equation is necessary to determine the potential φ. A commonly employed equation for this purpose is the Poisson-type equation, as described by Sharland et al. [29]. Thus, the potential can be obtained by solving Poisson's equation

$$\nabla^2\varphi = \frac{\sigma}{\varepsilon} \tag{2.44}$$

where ε is the permittivity of the electrolyte and σ is the charge density. In situations where the charge density is considerably smaller than the electric permittivity of the electrolyte, this equation simplifies to the electro-neutrality equation, and Poisson's equation is substituted with the local charge neutrality equation, which is expressed as [28, 29]

$$\sum_j z_j c_j = 0. \tag{2.45}$$

2.3.2.4 Rate of corrosion at the pits

Faraday's law of electrolysis determines the corrosion rate by establishing a connection between the electric charge and the amount of substance (m) involved in the electrochemical reaction [25]. Thus, the mass of the substance deposited or removed is given by:

$$\begin{aligned} m &\propto Q \\ &= \frac{MQ}{zF} \end{aligned}$$

where Q is the total amount of charge passed in t seconds, z is the number of electrons participating in the redox reaction, and M is the molar mass or molecular weight (g/mol). Thus the total amount of charge is obtained from the product of current (I) and time (i.e., $Q = It$), and inserting into the above equation gives:

$$\begin{aligned} m &= \frac{MIt}{zF} \\ &= \frac{MA}{zF} it, \end{aligned} \tag{2.46}$$

where i is the current density which represents the amount of electric current flowing through a specific area, $i = \frac{I}{A}$. If equation (2.46) is divided by the density ρ of the electrode material and the geometric surface area A, the loss per unit time t gives the corrosion penetration rate (CPR), v_{cpr}.

$$\begin{aligned} v_{cpr} &= \frac{M}{\rho z F} i \\ &= \frac{i}{zFc}, \end{aligned} \tag{2.47}$$

where c is the concentration and is related to the molar mass and density by $c = \frac{\rho}{M}$.

2.3.3 Crystallography for Corrosion

2.3.3.1 Crystal structure and unit cell

Crystal structure refers to the arrangement of atoms, ions, or molecules in a crystalline material. Crystals are solids with a regular and repeating three-dimensional pattern known as a lattice as demonstrated in Figure 2.9. This lattice structure is responsible for the characteristic geometric shapes and symmetries observed in crystals. The importance of crystal structure to various scientific disciplines and industries is due to its significant impact on material properties and behavior. For example, the arrangement of atoms in a crystal lattice directly influences material properties such as mechanical strength, thermal conductivity, electrical conductivity, and optical behavior, etc, [6]. Understanding crystal structure is essential for designing and tailoring materials with specific properties for various applications.

A unit cell is the basic building block of a crystal lattice, representing the smallest repeating unit of a crystal structure as represented in Figure 2.9. It is a three-dimensional parallelepiped (a six-faced figure with each face being a parallelogram) with edges defined by lattice parameters a, b, and c, and interaxial angles α, β, and γ.

There are several types of unit cells, with the most common for metals being the primitive cubic, body-centered cubic, and face-centered cubic unit cells, which represent different crystal structures based on the arrangement of atoms within the unit

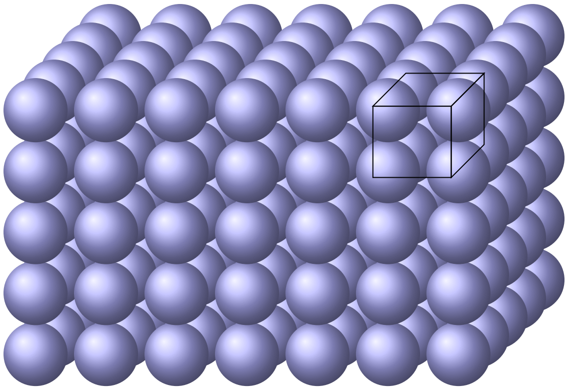


Figure 2.9: Crystal lattice of a simple cubic material with some atomic layers and a unit cell displayed.

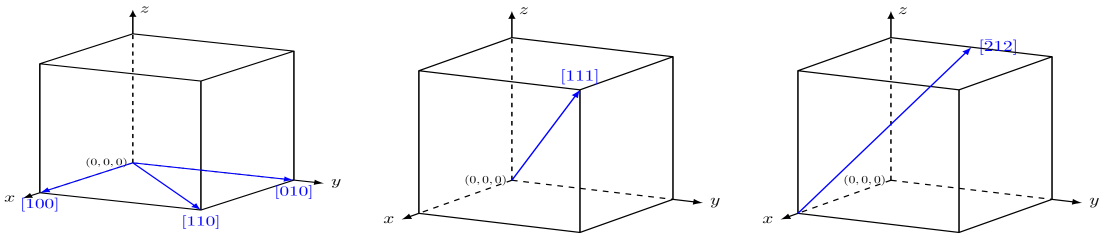
cell. The unit cell allows scientists and researchers to predict and analyze the physical, mechanical, and thermal properties of crystals, as well as predict their diffraction patterns using techniques like X-ray crystallography.

2.3.3.2 Miller indices, directions and planes

Miller indices represent the three-dimensional coordinate system for crystals, based on their unit cell. The indices are used in crystallography to describe the orientation of crystal planes and directions within a crystal lattice. The indices are represented by three integers denoted by $[h\ k\ l]$. Note that the values of Miller indices cannot be fractions, and there is no need for commas to separate the values of Miller indices.

In crystallography, square brackets [ ] represent a particular direction and angle brackets <> denote a family of directions. Parentheses () denote a specific plane, and curly brackets {} indicate a family of symmetrically equivalent planes. Miller indices use a unique notation for negatives, represented by a "bar" symbol (e.g., $\bar{1}$ for -1). For example, $[0\ \bar{1}\ 0]$ is pronounced as "the zero, bar one, zero direction."

In a cubic system, [1 0 0] and [0 1 0] are perpendicular directions and $< 1\ 0\ 0 >$ includes the directions $[1\ 0\ 0], [\bar{1}\ 0\ 0], [0\ 1\ 0], [0\ \bar{1}\ 0], [0\ 0\ 1]$, and $[0\ 0\ \bar{1}]$. The example of a direction for a face diagonal is [1 1 0] as shown in Figure 2.10(a), and [1 1 1] represents body diagonal for a cubic unit cell as displayed in Figure 2.10(b).

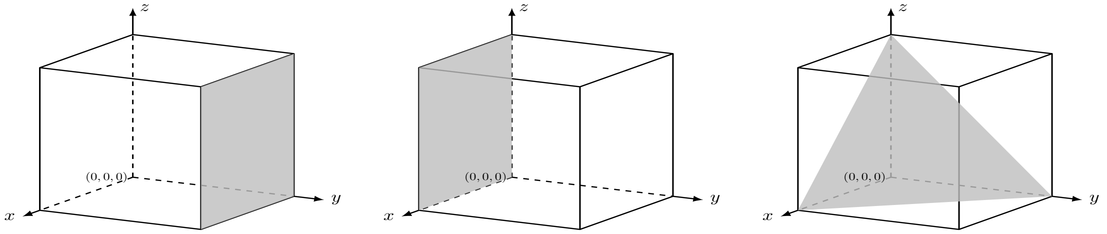


(a) (b) (c)

Figure 2.10: Miller indices of some directions within a cubic crystal.

Miller indices are defined with respect to a coordinate system so the first definition we need is an origin. The origin is a point (0,0,0) and we can define it anywhere in the crystal since the crystal is assumed to be infinite in all direction. For a given crystal direction, let's take the example of $[\bar{2}\ 1\ 2]$, we can draw the direction vector in the unit cell using Miller indices. It is important to note that Miller indices of directions always translate the direction so that it starts at the origin. The drawing procedure is as follows: first, determine the origin as the tail of the direction vector (if there is a negative number in the direction vector, move the origin in the positive direction for the axis). Then, determine the vector endpoint (remove a common factor if any of the indices are larger than 1). Finally, draw the vector. For our example: we start by pulling the common factor 2 out, i.e., $[\bar{2}12] = 2[\bar{1}\frac{1}{2}1]$. Then we draw the direction as shown in Figure 2.10(c).

Consider a unit cubic cell atoms at each corner. Now let us assume an imaginary plane passing through the unit cell (or lattice points of the unit cell). Miller indices



(a) Lattice plane (0 1 0). (b) Lattice plane at origin. (c) Lattice plane (1 1 1).

Figure 2.11: Lattice Planes.

for the plane are calculated by determining the intercepts made by the plane on the x, y, and z axes, respectively, and then take the reciprocals of these intercepts. If the plane passes through the origin, in this situation we need to shift the origin to the nearest lattice point on the parallel face.

Example 2.3.1. A plane which perpendicular to y-axis at $y = 1$ as shown in Figure 2.11(a). . Since the reciprocals of x, y and z intercepts are 0, 1, and 0 respectively, the Miller indices of this plane is (0 1 0).

Example 2.3.2. In Figure 2.11(b), since the plane passes through the origin, we shift the origin to the nearest lattice point on the parallel face. Then the reciprocals of x-intercept, y-intercept and z-intercept are 0, -1, and 0 respectively. Therefore the Miller indices of this plane are $(0\ \bar{1}\ 0)$.

Example 2.3.3. The reciprocals of the x, y and z intercepts are 1, 1 and 1, respectively for the plane shown in Figure 2.11(c). Therefore, the Miller indices of this plane are (1 1 1).

Why are the crystal directions and planes important? They provide essential information about the arrangement and symmetry of atoms within a crystal lattice, which profoundly impacts the material's properties and behavior. In our work, crystal directions and planes plays an important role in the calculation of the corrosion potential for a non-homogeneous material.

2.3.3.3 Relation between crystal coordinate and Cartesian coordinates systems using a rotation matrix

The relation between crystal coordinates and Cartesian coordinate systems is essential because it allows us to describe the position and orientation of crystallographic features within a crystal in terms of familiar three-dimensional Cartesian coordinates. This relationship is crucial for understanding and analyzing the crystal's physical properties and behavior.

Let D represent the original set of basis vectors for cubic crystal,

$$D = \{[100], [010], [001]\}.$$

In a cubic crystal system, the basis vectors are perpendicular to each other. Now, let us define a generalized set of vectors for a given crystal direction $[h\ k\ l]$:

$$C = \{\hat{\imath}, \hat{\jmath}, [h\ k\ l]\}$$

where $\hat{\jmath}$ can be found from the relation $\hat{\jmath} = [h\ k\ l] \times \hat{\imath}$, and we just need to set the vector $\hat{\imath}$. This turns out to be easy for a cubic crystal system since the equation for a dot product is straightforward. Thus, any vector that satisfies:

$$\hat{\imath} \cdot [h\ k\ l] = 0 \implies i_x h + i_y k + i_z l = 0.$$

where i_x, i_y and i_z can be the components of the vector $\hat{\imath}$.

Now let us look at an example. Assume we have $[1\ 1\ 1]$ coming out of the page (defined as the zone axis of the crystal), $h = 1$, $k = 1$, $l = 1$. Then we get:

$$i_x + i_y + i_z = 0$$

Since there are two independent variables, we can choose $i_z = 0$ for simplicity, thus that the vector lies in the xy-plane (no component out of the plane). The simplest choice is $i_x = 1$, and we get:

$$1 + i_y + 0 = 0 \implies i_y = -1$$

This gives us $\hat{\imath} = [1\ -1\ 0]$ (we neglect the normalizing factor), and

$$\begin{aligned}
\hat{\jmath} &= \hat{k} \times \hat{\imath} \\
&= \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ 1 & 1 & 1 \\ 1 & -1 & 0 \end{vmatrix} \\
&= [\hat{x}\ \hat{y}\ \hat{z}] \\
&= [1\ 1\ -2]
\end{aligned}$$

where $\hat{x}$, $\hat{y}$ and $\hat{z}$ are the unit vectors. Hence the matrix which converts vectors from one basis to another is [26]:

$$
\begin{aligned}
\mathbf{A}^{-1} &= \left[\begin{array}{c|c|c} & & \\ \hat{i} & \hat{j} & \hat{k} \\ & & \end{array}\right], \\
&= \left[\begin{array}{c|c|c} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -2 & 1 \end{array}\right].
\end{aligned}
$$

Now the question is how do we use this matrix? Let $\mathbf{n}_{CD}$ be the outward unit normal in the crystal coordinate system, and using the notation given in [4, 26], its relationship to unit normal $\mathbf{n}$ in the Cartesian coordinate system is

$$\mathbf{n}_{CD} = \mathbf{A}^{-1}\mathbf{n}. \tag{2.48}$$

For our [1 1 1] example, we have

$$
\begin{aligned}
\mathbf{n}_{CD} &= \left[\begin{array}{c|c|c} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -2 & 1 \end{array}\right] \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \\
&= \begin{bmatrix} n_x + n_y + n_z \\ -n_x + n_y + n_z \\ -2n_y + n_z \end{bmatrix}.
\end{aligned}
$$

Table 2.2 displays the relation between $\mathbf{n}$ and $\mathbf{n}_{CD}$. Hence $\mathbf{A}^{-1}$ matrix is a transfor-

$\mathbf{n}$	$\mathbf{n}_{CD}$
[1 0 0]	$[1\ \bar{1}\ 1]$
[0 1 0]	$[1\ 1\ \bar{2}]$
[0 0 1]	[1 1 1]

Table 2.2: The values of $\mathbf{n}$ and $\mathbf{n}_{CD}$ for a zone axis of [1 1 1] .

mation matrix which can be calculated for any crystal orientation defined normal to the plane of the given crystal. The $\hat{\imath}$ vector can be chosen in any crystalline direction, therefore the crystal is rotated about [0 0 1] any way we wish.

2.3.4 Modeling approach of pitting corrosion

The computational models for pitting corrosion can be divided into two principal categories based on how they compute the corrosion front: non-autonomous and autonomous models. In non-autonomous models, numerical methods are used to solve the transport equation separately for the motion of the corrosion front and the evolution of the pit domain. On the other hand, autonomous models describe the dissolution or transport mechanism together with the pit propagation. Since non-autonomous models are a well-established framework and commercially used [18], we have chosen the non-autonomous model for this thesis work. Particularly, non-autonomous models use the FEM and track the evolution of pit growth using a moving boundary technique. The velocity vector (2.47) is computed at each boundary node for pit propagation.

The thesis does not focus on atomistic models for corrosion processes that study chemical reactions in detail. Instead, we focus on a mathematical model and solve the model using the adaptive moving mesh approach. If the reader is interested in atomistic models or related matters, please refer to articles [23, 30, 31, 32].

Most researchers choose micrometer to millimeter length scales for better understanding the initial formation and propagation of corrosion pits. All simulations in this thesis are computed at the micrometer scale.

Bibliography

[1] KV Akpanyung and RT Loto. Pitting corrosion evaluation: A review. In *Journal of Physics: Conference Series*, volume 1378, page 022088. IOP Publishing, 2019.

[2] Hassan Abdul-Zehra Abdul-Yima Al-Fetlawi. *Modelling and Simulation of All-vanadium Redox Flow Batteries.* PhD thesis, University of Southampton, 2011.

[3] Allen J Bard, Larry R Faulkner, Johna Leddy, and Cynthia G Zoski. *Electrochemical Methods: Fundamentals and Applications*, volume 2. Wiley New York, 1980.

[4] Monte B Boisen and Gerald V Gibbs. *Mathematical Crystallography*, volume 15. Walter de Gruyter GmbH & Co KG, 2018.

[5] Theodore L Brown, H Eugene LeMay, and Bruce Edward Bursten. *Chemistry: the Central Science.* Pearson Educación, 2002.

[6] Sayantan Chattoraj, Limin Shi, and Changquan Calvin Sun. Understanding the relationship between crystal structure, plasticity and compaction behaviour of theophylline, methyl gallate, and their 1: 1 co-crystal. *CrystEngComm*, 12(8):2466–2472, 2010.

[7] Carl de Boor. Good approximation by splines with variable knots. In *Spline Functions and Approximation Theory*, pages 57–72. Springer, 1973.

[8] Virginia G DeGiorgi, Nithyanand Kota, Alexis C Lewis, and Siddiq M Qidwai. Numerical modeling of pit growth in microstructure. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55850, pages 1–7. American Society of Mechanical Engineers, 2013.

[9] Kelsey L DiPietro, Ronald D Haynes, Weizhang Huang, Alan E Lindsay, and Yufei Yu. Moving mesh simulation of contact sets in two dimensional models of elastic–electrostatic deflection problems. *Journal of Computational Physics*, 375:763–782, 2018.

[10] David Scott Dodson. *Optimal-Order Approximation by Polynomial Spline Functions.* Purdue University, 1972.

[11] Adolf Fick. Ueber diffusion. *Annalen der Physik*, 170(1):59–86, 1855.

[12] Mars G Fontana. Corrosion engineering, 1987.

[13] GS Frankel. Pitting corrosion. *ASM handbook*, 13:236–241, 2003.

[14] Godfrey Harold Hardy, John Edensor Littlewood, György Pólya, DE Littlewood, G Polya, et al. *Inequalities*. Cambridge University Press, 1952.

[15] Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *Journal of Computational Physics*, 174(2):903–924, 2001.

[16] Weizhang Huang. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics*, 204(2):633–665, 2005.

[17] Weizhang Huang and Robert D Russell. *Adaptive Moving Mesh Methods*, volume 174. Springer Science & Business Media, 2010.

[18] Siavash Jafarzadeh, Ziguang Chen, and Florin Bobaru. Computational modeling of pitting corrosion. *Corrosion Reviews*, 37(5):419–439, 2019.

[19] DA Jones. The technology and evaluation of corrosion. *Principles and Prevention of Corrosion, 2nd ed.; Prentice-Hall: Upper Saddle River, NJ, USA*, 1996.

[20] Denny A Jones. Principles and prevention. *Corrosion*, 2:168, 1996.

[21] Nithyanand Kota, Siddiq M Qidwai, Alexis C Lewis, and Virginia G DeGiorgi. Microstructure-based numerical modeling of pitting corrosion in 316 stainless steel. *ECS Transactions*, 50(31):155–164, 2013.

[22] Philippe Marcus. *Corrosion Mechanisms in Theory and Practice*. CRC press, 2011.

[23] Philippe Marcus and Vincent Maurice. Atomic level characterization in corrosion studies. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2098):20160414, 2017.

[24] Howard W Pickering. The role of electrode potential distribution in corrosion processes. *Materials Science and Engineering: A*, 198(1-2):213–223, 1995.

[25] Branko N Popov. *Corrosion engineering: principles and solved problems*. Elsevier, 2015.

[26] M. Robertson and K. Raffel. Imaging crystals. *Microscopical Society of Canada Bulletin*, 36:13–18, 08 2008.

[27] Suhaila Salleh. *Modelling Pitting Corrosion in Carbon Steel Materials*. The University of Manchester, Manchester, United Kingdom, 2013.

[28] SM Sharland, CP Jackson, and AJ Diver. A finite-element model of the propagation of corrosion crevices and pits. *Corrosion Science*, 29(9):1149–1166, 1989.

[29] SM Sharland and PW Tasker. A mathematical model of crevice and pitting corrosion-I. The physical model. *Corrosion Science*, 28(6):603–620, 1988.

[30] Christopher D Taylor. Modeling corrosion, atom by atom. *The Electrochemical Society Interface*, 23(4):59, 2014.

[31] Christopher D Taylor and Philippe Marcus. *Molecular Modeling of Corrosion Processes: Scientific Development and Engineering Applications*. Wiley Online Library, 2015.

[32] Chandrabhan Verma, H Lgaz, DK Verma, Eno E Ebenso, I Bahadur, and MA Quraishi. Molecular dynamics and monte carlo simulations as powerful tools for study of interfacial adsorption behavior of corrosion inhibitors in aqueous phase: a review. *Journal of Molecular Liquids*, 260:99–120, 2018.

[33] Andrew B White, Jr. On selection of equidistributing meshes for two-point boundary-value problems. *SIAM Journal on Numerical Analysis*, 16(3):472–502, 1979.

[34] Lietai Yang. *Techniques for corrosion monitoring*. Woodhead Publishing, 2020.

Chapter 3

Moving mesh simulation of pitting corrosion

Abstract

[1] Damages due to pitting corrosion of metals cost industry billions of dollars per year and can put human lives at risk. The design and implementation of an adaptive moving mesh method is provided for a moving boundary problem related to pitting corrosion. The adaptive mesh is generated automatically by solving a mesh PDE coupled to the nonlinear potential problem. The moving mesh approach is shown to enable initial mesh generation, provide mesh recovery and is able to smoothly tackle changing pit geometry. Materials with varying crystallography are considered. Changing mesh topology due to the merging of pits is also handled. The evolution of the pit shape, the pit depth and the pit width are computed and compared to existing results in the literature.

Keywords: Pitting Corrosion, Adaptive Moving Mesh, MMPDE, FEM, Crystallography

Declaration of Competing Interests: We have no competing interests to declare.

[1] This work has been accepted as "*Moving mesh simulation of pitting corrosion*", by A. N. Sarker, R. D. Haynes, and M. D. Robertson, to the Journal of Mathematics in Science and Industry, 2023.

3.1 Introduction

Corrosion is a deterioration or breakdown of a material due to chemical or electrochemical reactions. In particular, pitting corrosion is one of the most disastrous and devastating localized forms of corrosion; generating a small pit, cavity or hole in the metal. Pitting corrosion is difficult to identify, and can have a big impact on the structural integrity of metal [47, 56]. The pit geometry depends on many factors such as the components of the metal, the surface orientation, and the physical and chemical environment at the time of attack [43]. Corrosion pits can have different shapes [34] and with the ability to grow over time, failure of engineering structures such as bridges, pipelines and nuclear power plants may result [9, 34, 39].

Computational modeling and simulations have been a tremendous asset in the study of pitting corrosion over a wide range of conditions and materials. Determining the pitting behavior experimentally is time consuming, expensive and physically difficult or impossible in many situations. Numerical simulations allow us to study pitting under a wide range of conditions within a reasonable time.

In last few decades, several review papers that have focused on partial differential equation (PDE) based models for pitting corrosion based on finite element or finite volume methods [10, 11, 31, 32, 42, 44, 46, 50]. In 2019, an extensive overview of the mathematical models for pitting corrosion based on the anodic reaction at the corrosion front, the transportation of ions in the pits of the electrolyte domain and the pit growth over time is provided in [26]. In many of these studies COMSOL® is used to solve the PDE in the electrolyte domain and the corrosion front movement and meshing is computed by the arbitrary Lagrangian-Eulerian (ALE) approach and the level set method [11, 30]. In other studies, a 2D PDE model is solved with the finite element method [13, 49] and the finite volume method [41, 42]. Pit growth is determined by finite element methods and a level set approach in [49], and using an extended finite element method (XFEM) and level set method in [13]. In 2020, an ALE method is implemented to move the mesh at the pit boundary and analyze the relationship between the corrosion behavior and the local corrosive environment within a single pit [51].

The previously mentioned FEM approaches relied on a complete remeshing of the domain at every time step. An alternative technique, presented here, uses an adaptive moving mesh method where the mesh size, shape and orientation of the mesh elements are automatically and continuously varied for each time step, while keeping the number of nodes and mesh topology fixed throughout the computation.

Continuous mesh movement approaches are divided into two main categories: velocity-based approaches and location-based approaches. Most velocity-based approaches are motivated by the Lagrangian algorithm, where the mesh movement is tightly associated with the fluid or material particle flow. The Eulerian approach has a fixed computational mesh and the continuum moves with respect to the mesh nodes. The Eulerian and Lagrangian algorithms are commonly used in fluid dynamics and structural material problems, respectively [12]. In general, Eulerian meshes avoid mesh tangling and diffusive solutions, but the method can have difficulty adjusting to sharp material interfaces. One of the advantages of the Lagrangian approach is that the advective terms do not appear in the governing equations. Thus, the Lagrangian methods are less diffusive compared to the Eulerian approach, while also maintaining sharp material interfaces [22]. The ALE methods are velocity-based methods, which provide a combination of Lagrangian and Eulerian approaches [14, 15, 29, 35, 37, 52].

The main goal of the location-based mesh movement approach is to directly control the location of mesh points in particular regions of the computational domain. A typical location-based method is the variational approach, which relocates the mesh points by movements that are based on minimizing a functional formulated to measure the difficulty or the error in the numerical solution [22]. Other location-based algorithms are based on elliptic PDE descriptions which can be used to generate boundary-fitted meshes [48, 53], sometimes known as Winslow's approach [54]. Winslow's idea can be generalized using a functional [6], which provides a combination of the mesh adaptivity, smoothness, and orthogonality conditions.

A number of articles consider mesh adaptation functionals including mechanical models [23, 24, 25], vector fields [27], a weighted Jacobian matrix [28, 29], a matrix-valued diffusion coefficient [8, 20], and the equidistribution and isotropy (or alignment conditions) presented in [16]. The moving mesh PDE (MMPDE) method that we use has been developed by several authors [7, 19, 19, 20, 21, 38]. Therein, the mesh movement is determined by a gradient flow equation, and the functional plays the vital role.

To our knowledge, an adaptive moving mesh method, our method of choice, has not been implemented for PDE-based modelling of pitting corrosion. In our moving mesh method, the FEM provides the spatial discretization and our computational framework is built upon the package MMPDElab by Huang [18]. MMPDElab is a general adaptive moving mesh finite element solver for time dependent PDEs based on integration of the MMPDE. The solver uses an alternating mesh and physical solution approach, and we used the solver to achieve an adaptive moving mesh which provides sufficient mesh elements in and around the pit. We focus on the development of an appropriate mesh density function which implicitly and automatically determines an appropriate distribution of nodes as the pit evolves. Our paper provides: 1) a proof of concept implementation of the moving mesh approach for pitting corrosion, 2) an adaptive solver for both single and multiple crystal directions, 3) the ability to handle single and multiple pits, and 4) the ability to provide (provably) nonsingular quality evolving meshes in an automatic way. The test material for demonstrating these techniques is 316 stainless steel using the parameters provided in [11, 30]. This paper is organized as follows. We discuss the preliminaries of the pitting corrosion mechanism, the crystal orientation, the PDE-model, and an overview of the moving mesh methodology in Section 3.2. The finite element method approach for the physical PDE, the choice of mesh density function and moving mesh parameters, initial mesh generation, the boundary movement strategy, and the overall alternating solution approach are discussed in Section 3.3. Section 3.4 is devoted to our numerical results and validation.

3.2 Model problem

In this section we detail our prototype model problem and adaptive solution strategy including the description of the domain, model PDEs and boundary conditions, the necessary crystallography, and an overview of the moving mesh strategy.

3.2.1 PDE model equation

The transport of species in the solution can be modeled by the conservation of mass [45, 55], which leads to the mathematical form

$$\underbrace{\frac{\partial c_j}{\partial t}}_{\text{Storage}} = \underbrace{-\nabla \cdot \mathbf{N}_j}_{\text{Flux in - Flux out}} + \underbrace{R_j}_{\text{Generation}}, \tag{3.1}$$

where t is time, c_j is the concentration of the jth species, $\mathbf{N}_j$ is the flux of the jth species, and R_j is the rate of species generation due to chemical reactions. The ionic flux depends on the gradient of ion concentration, electro-migration and convection (or flow in a liquid medium). For each individual species i, the transport of the species in the electrolyte is described by the Nernst-Planck equation as

$$\mathbf{N}_j = -D_j \nabla c_j - z_j F u_j c_j \nabla \varphi + c_j \mathbf{v}, \tag{3.2}$$

where D_j is the diffusion coefficient of the jth species, u_j is the mobility of the species, z_j is the charge of the species, φ is the electric potential, F is Faraday's constant, and $\mathbf{v}$ is the solvent velocity [2]. Equation (3.2) gives the flux of the species as a combination of three contributing fluxes. The term $-D_j \nabla c_j$ describes the diffusive flux, the term $-z_j F u_j c_j \nabla \varphi$ gives the electro-migration flux and the term $c_j \mathbf{v}$ is the convection flux. Generally, the diffusion coefficient varies with the position of the species but we assume the diffusion coefficient is constant in our model.

While equation (3.2) represents the complete coupled behavior observed in the electrolyte domain during pitting corrosion, certain physically motivated assumptions are made in arriving at our simplified model: a) the absence of gradients in the species concentration due to the rapid mixing of the electrolyte; b) the incompressibility of the solvent; c) the zero net production of the reactants; and d) the electro-neutrality of the electrolyte solution. These assumptions simplify (3.1) to the well-known Laplace equation. In this case, the electrolyte potential can be found by solving (on the electrolyte domain, Ω, shown in Figure 3.1)

$$\nabla^2 \varphi = 0 \text{ in } \Omega, \tag{3.3}$$

with the following boundary conditions

$$\begin{aligned} \varphi &= 0 \text{ on } \Gamma_1, \\ \nabla \varphi \cdot \mathbf{n} &= 0 \text{ on } \Gamma_2, \Gamma_3, \Gamma_4, \\ \nabla \varphi \cdot \mathbf{n} &= \frac{i_a(\varphi)}{\sigma_c} \text{ on } \Gamma_p, \end{aligned} \tag{3.4}$$

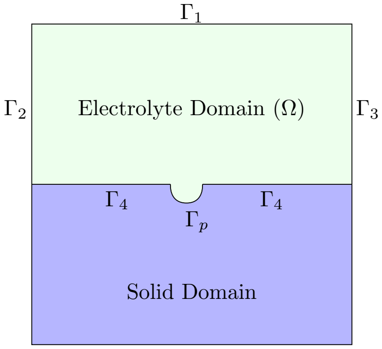


Figure 3.1: The 2D computational domain.

where $\nabla \varphi \cdot \mathbf{n} = \frac{\partial \varphi}{\partial \mathbf{n}}$, $\mathbf{n}$ is the (outward) unit normal vector with respect to Ω, $i_a(\varphi)$ is the anodic current density, σ_c is the electrical conductivity of the electrolyte, Γ_p is the pit boundary, and Γ_1, Γ_2, Γ_3 and Γ_4 are the top, left, right and bottom of the domain (excluding the pit boundaries), respectively. The boundary condition $\frac{\partial \varphi}{\partial \mathbf{n}} = 0$ enforced on Γ_2, Γ_3 and Γ_4 ensures there is no flow of ions across these boundaries. We denote the horizontal and vertical co-ordinates of the electrolyte region in Figure 3.1 by x and y, respectively.

The current density is modelled by the Butler-Volmer relation

$$i_a(\varphi) = zFA_s \cdot e^{\left(\frac{zF(V_{corr}+\alpha\eta_a)}{RT}\right)}, \tag{3.5}$$

where α is the transfer coefficient, z is the average charge number for the dissolving metal, A_S is the material dissolution affinity, T is the temperature and R is the universal gas constant [3].

The Butler-Volmer relation is used to describe the experimental data as a function of the applied over-potential

$$\eta_a = V_{app} - V_{corr} - \varphi,$$

where V_{app} and V_{corr} are the applied and the corrosion potentials, respectively [42].

Parameter	:	**Description**	**Value**
z	:	Average charge number for the metal	2.19
F	:	Faraday's constant	96485 C/mol
R	:	Universal gas constant	8.315 J/(mol K)
T	:	Temperature	298.15 K
V_{corr}	:	Mean corrosion potential (homogeneous)	-0.24 V
V_{app}	:	Applied potential	-0.14 V
A_{diss}	:	Dissolution affinity	4 mol/cm^{2}s
C_{solid}	:	Solid concentration	143 mol/l
Δt	:	Time step size	1

Table 3.1: List of parameters used in the corrosion model.

As the metal corrodes, the pit boundary moves as the pit becomes larger. In our model the new position of corrosion front, X_{new}, is computed from the old position, X_{old}, by a simple time stepping procedure

$$X_{\text{new}} = X_{\text{old}} + \Delta t V_n \mathbf{n},$$

where V_n is the magnitude of normal velocity. The magnitude of normal velocity at the corrosion interface (or the movement of the corrosion front) is described using Faraday's law

$$V_n = \frac{i_a(\varphi)}{zFc_{solid}}, \tag{3.6}$$

where c_{solid} is the atomic mass concentration of the metal and z is the average charge number for the metal. Table 3.1 gives a list of parameters.

3.2.2 Crystal orientation and corrosion potential

The corrosion potential is the mathematical link between the etching effects of the electrolyte and the material undergoing pitting. This potential term is present in the Butler-Volmer equation (3.5) and is an important parameter governing the velocity of the sides of the pit during corrosion. If the electrolyte etches the material homogeneously in all directions then the crystal structure is not a variable in the modeling and only one number is required for V_{corr}. However, if the material is crystalline then the corrosion potential may vary dependent on the particular crystallographic surface exposed to the electrolyte. Hence, a connection between the Cartesian $(x, y, 0)$ geometry used for defining the computational domains as presented in Figure 3.1 and the directions in the crystal, is needed. In general, these two geometries will not align since crystals can be rotated to lie along an infinite number of directions and a transformation will be required to relate the two coordinate systems. Letting $\mathbf{n}_{CD}$ represent the outward unit normal in the crystal coordinate system, and using the notation given in [5, 40], its relationship to $\mathbf{n}$ is

$$\mathbf{n}_{CD} = \mathbf{A}^{-1}\mathbf{n}. \tag{3.7}$$

The matrix $\mathbf{A}^{-1}$ is defined by

$$\mathbf{A}^{-1} = \left[\begin{array}{c|c|c} \mathbf{i} & \mathbf{j} & \mathbf{k} \end{array} \right], \tag{3.8}$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are orthogonal unit column vectors in Cartesian space. $\mathbf{k}$ is chosen as the desired zone axis of the crystal and $\mathbf{i}$ is chosen as the direction perpendicular to $\mathbf{k}$ which will be oriented along the x axis in the computational domain. Thus, the crystal can be rotated in any direction about the zone axis offering maximum flexibility in the problems that can be studied. The third unit vector, $\mathbf{j}$, is found using the perpendicularity property of vector cross products

$$\mathbf{j} = \mathbf{k} \times \mathbf{i}.$$

In order to better see how to form the $\mathbf{A}^{-1}$ matrix, two examples will be presented. First, select a crystal orientation where the zone axis is aligned along $\mathbf{k} = [001]$. Next, choose the $\mathbf{i} = [100]$ direction to be along the x computational domain direction. Performing the $\mathbf{j} = \mathbf{k} \times \mathbf{i}$ cross product, it is found that $\mathbf{j} = [010]$. Hence, the transformation matrix is

$$\mathbf{A}_{001}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In this case, the transformation matrix is the identity matrix and the crystal coordinate system is the same as the computational domain coordinate system, ie. $\mathbf{n}_{CD} = \mathbf{n}$. For the second example, choose the zone axis to be along the [101] direction and the $[\bar{1}01]$ crystal direction to be along the x computational domain direction. This leads to $\mathbf{j} = [010]$. After normalizing the vectors, the transformation matrix in this case is

$$\mathbf{A}_{110}^{-1} = \begin{bmatrix} \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

The next step is to define a corrosion potential for each crystallographic direction. Unfortunately, experimental data giving the corrosion potential as a function of crystallographic surface is usually not available, hence we will adopt a similar form of semi-empirical potential for 316 stainless steel

$$V_{corr} = k - s\left[1 - (\langle 001 \rangle \cdot \mathbf{n}_{CD})_{\max}\right], \tag{3.9}$$

where $k = -0.2297$ and $s = 0.054$ gives a 10% difference between the maximum and minimum V_{corr} values, that is between the [001] and [111] crystal planes as used by DeGiorgi et al. in [30]. We write $(\langle 001 \rangle \cdot \mathbf{n}_{CD})_{\max}$, rather than $[001] \cdot \mathbf{n}_{CD}$ as used in [30]. In our case, $\langle 001 \rangle$ represents any one of the six cryptographically equivalent [001], $[\bar{1}00]$, [010], $[0\bar{1}0]$, [001], $[00\bar{1}]$ directions that maximizes the dot product with the crystal direction normal vector. Maximizing this dot product minimizes the angle between the normal vector and the particular $\langle 001 \rangle$ vector so that an equivalent result to the standard stereographic triangle is obtained. Figure 3.2 is an example of this procedure is provided in where a single crystal has been oriented along the [001] zone axis. The semicircle represents the pit boundary, the black lines and arrows the outward pit normal vectors, and the heavy blue line is the value of V_{corr} as a function of position along the edge of the pit. Highlighted by the three colours are sections along the pit boundary that have a different ⟨001⟩ vector for use in equation (3.9). These vectors are presented in blue text and are [001], [0$\bar{1}$0] and [$\bar{1}$00] for the right, centre and left sections of the pit, respectively.

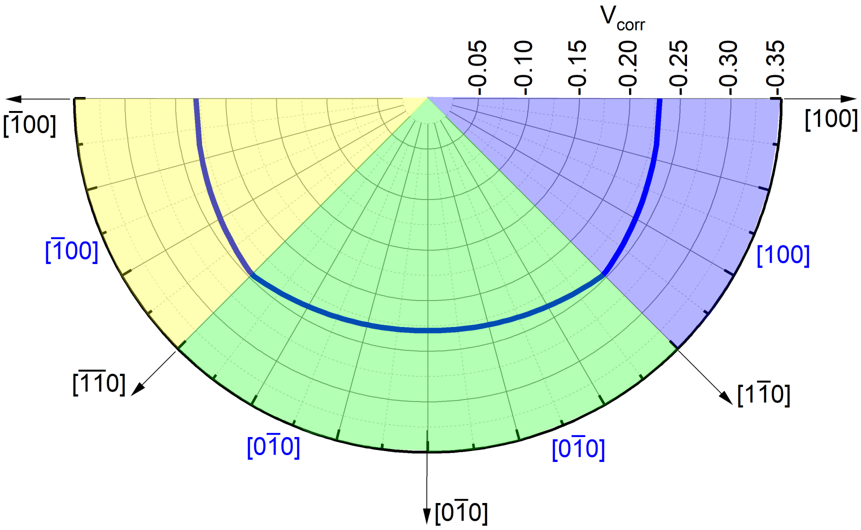


Figure 3.2: A plot of V_{corr} as a function of location around the pit boundary. The blue, green and yellow sections of the pit edge require different $< 001 >$ vectors for use in equation (3.9).

3.2.3 Overview of the moving mesh strategy

The basic idea of the moving mesh method is to automatically redistribute a fixed number of nodes where additional accuracy is required. The mesh moves or evolves automatically as the solution or domain evolves and is obtained by solving a MMPDE. This MMPDE depends on a mesh density or monitor function, which is large where the mesh density is needed to be large. The mesh density function is often chosen to depend on variations or errors in the solution of the physical PDE or is chosen by geometrical considerations (as in this paper).

In 1D, the equidistribution principle (EP) is used to derive the moving mesh system. The EP uses a mesh density function $\boldsymbol{\rho} = \boldsymbol{\rho}(x) > 0$ which is to be distributed evenly among the mesh elements in the domain. Given an integer $N > 1$, the continuous and bounded function $\boldsymbol{\rho}$ on $[a, b]$ is evenly distributed on the mesh $\mathcal{T}_h = a = x_0 < x_1 < \cdots < x_N = b$, if

$$\int_{x_0}^{x_1} \boldsymbol{\rho}(x)dx = \int_{x_1}^{x_2} \boldsymbol{\rho}(x)dx = \cdots = \int_{x_{N-1}}^{x_N} \boldsymbol{\rho}(x)dx. \tag{3.10}$$

A mesh $\mathcal{T}_h$ is called an equidistributing mesh if the mesh satisfies the equidistribution principle.

The physical problem is assumed to require a non-uniform x-coordinate, $x \in \Omega$. This physical coordinate, x, is a mapping of the computational ξ-coordinate where $\xi \in \Omega_c = [0, 1]$, and $x(0) = a$ and $x(1) = b$, if $\Omega = [a, b]$. We attempt to generate a physical mesh $\mathcal{T}_h$ using a mesh transformation $x = x(\xi) : \Omega_c \to \Omega$ and a uniform mesh in the ξ-coordinate

$$\xi_i = \frac{i}{N}, \quad i = 0, 1, \ldots, N.$$

The equidistribution principle (3.10) can then be written as

$$\begin{aligned} \int_a^{x_i} \boldsymbol{\rho}(\tilde{x})d\tilde{x} &= \frac{i}{N}\int_a^b \boldsymbol{\rho}(\tilde{x})d\tilde{x} \\ &= \frac{i}{N}\boldsymbol{\sigma}, \quad i = 0, 1, \ldots, N, \end{aligned}$$

where $\boldsymbol{\sigma} = \int_a^b \boldsymbol{\rho}(\tilde{x})d\tilde{x}$. The function $\int_a^x \boldsymbol{\rho}(\tilde{x})d\tilde{x}$ is strictly monotonically increasing if $\boldsymbol{\rho} > 0$, therefore each x_i is unique. Here $\boldsymbol{\sigma}$ and $\frac{i}{N}\boldsymbol{\sigma}$ are the total error and average error in the approximating solution, respectively.

Using the mesh transformation we have

$$\int_a^{x(\xi_i)} \boldsymbol{\rho}(\tilde{x})d\tilde{x} = \xi_i\boldsymbol{\sigma}, \quad i = 0, 1, \ldots, N,$$

and the continuous version is given by

$$\int_a^{x(\xi)} \boldsymbol{\rho}(\tilde{x})d\tilde{x} = \xi\boldsymbol{\sigma}, \quad \forall\xi \in \Omega_c. \tag{3.11}$$

The continuous mapping $x = x(\xi)$ is called a *equidistributing coordinate transformation* for $\boldsymbol{\rho}$ if it satisfies relation (3.11). Differentiating (3.11) with respect to ξ gives

$$\boldsymbol{\rho}(x)\frac{dx}{d\xi} = \boldsymbol{\sigma}. \tag{3.12}$$

Equation (3.12) indicates that $\frac{dx}{d\xi}$ is small when $\boldsymbol{\rho}$ is large. Again, differentiating with respect to ξ gives

$$\frac{d}{d\xi}\Big(\boldsymbol{\rho}(x)\frac{dx}{d\xi}\Big) = 0, \tag{3.13}$$

with the boundary conditions

$$x(0) = a, \quad x(1) = b. \tag{3.14}$$

This is a nonlinear boundary value problem (BVP) for the required mesh transformation and physical mesh. The mesh and physical solution on that mesh is determined by solving this BVP and the physical PDE as a coupled system.

In higher dimensions, in order to describe the equidistribution and alignment conditions at the discrete level, we consider a mesh $\mathcal{T}_h$ of N triangular elements with N_v vertices in the physical domain $\Omega \in \mathbf{R}^d$ $(d \geq 1)$. Furthermore, we consider an invertible affine mapping $F_K : \hat{K} \to K$ and its Jacobian matrix, $F_K^{'}$, where $\hat{K}$ is the reference or master element for a physical element K in $\mathcal{T}_h$. Assume that a metric tensor (or a monitor function) $\mathbb{M} = \mathbb{M}(x)$ is given on Ω which determines the shape, size and orientation of mesh elements of the domain Ω. Generally, a mesh is uniform if all of its elements have the same size and is similar to a reference element $\hat{K}$. So, the main idea of the MMPDE method is to view any adaptive mesh $\mathcal{T}_h$ as a uniform mesh in the metric $\mathbb{M}$.

The requirements of the equidistribution and alignment in higher dimensions can be expressed mathematically at the discrete level [22] as

$$|K|\sqrt{\mathtt{det}(\mathbb{M}_K)} = \frac{\sigma_h}{N}, \forall K \in \mathcal{T}_h,$$

and

$$\frac{1}{n}tr((F_K^{'})^T\mathbb{M}_K F_K^{'}) = \mathtt{det}((F_K^{'})^T\mathbb{M}_K F_K^{'})^{\frac{1}{d}}, \forall k \in \mathcal{T}_h,$$

where $|K|$ is the volume of K and $\sigma = \sum_{K\in\mathcal{T}_h}|K|\sqrt{\mathtt{det}(\mathbb{M}_k)}$.

A standard choice bases the metric tensor $\mathbb{M}$ on the approximate Hessian of the solution. This choice of $\mathbb{M}$ is known to be optimal with respect to the L_2 norm of the linear interpolation error [17]. Here we focus on the evolution of the pit geometry and choose $\mathbb{M}$ through geometrical considerations (see Section 3.3.2).

A discrete functional associated with the equidistribution and alignment conditions is given by

$$I[\mathcal{T}_h] = \sum_{K\in\mathcal{T}_h} |K|\sqrt{\det(\mathbb{M}_K)}\left[\theta\left(\mathrm{tr}(\mathbb{J}\mathbb{M}_K^{-1}\mathbb{J}^T)\right)^{\frac{d\gamma}{2}} + (1-2\theta)d^{\frac{d\gamma}{2}}\left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}_K)}}\right)^{\gamma}\right], \quad (3.15)$$

where $\mathbb{J} = (F_K^{'})^{-1}$. Minimizing this functional $I[\mathcal{T}_h]$ approximately satisfies the equidistribution and alignment conditions [16]. The value of the parameters $\theta = \frac{1}{3}$, and $\gamma = \frac{3}{2}$ are used for our numerical experiments.

The MMPDE moving mesh equation can then be defined as the (modified) gradient system (or gradient flow equation) for the energy functional, i.e.,

$$\frac{d\mathbf{x}_i}{dt} = -\frac{P_i}{\tau}\frac{\partial I[\mathcal{T}_h]}{\partial \mathbf{x}_i}, \quad i = 1, 2, \ldots, N_v, \quad t \in (t_n, t_{n+1}], \quad (3.16)$$

where $P_i = \det(\mathbb{M}_i)^{\frac{1}{d+2}}$ is a scalar function used to ensure the mesh equation has invariance properties and τ is a positive parameter used to adjust the response time of mesh movement to the change in $\mathbb{M}$. A smaller value of τ provides a faster response.

3.3 The numerical implementation

This section describes the details of the adaptive MMPDE strategy used to solve the PDE pitting corrosion model using a customized version of MMPDELab [18].

3.3.1 Discretization of the physical PDE

MMPDElab requires the user to specify the physical PDE in weak form, where the strong form of our model problem is given in equations (3.3) and (3.4). Let V be the trial space, chosen in this case as

$$V = \{v \in H^1(\Omega(t)) : v = 0 \text{ on } \Gamma_1\} \subset H^1(\Omega(t)),$$

where $H^1(\Omega(t))$ is, roughly speaking, the function space whose members, and their first derivatives, are square integrable (see, for example, [1] for details). At any time t the weak form is constructed as follows: find $\varphi \in V$ such that

$$\int_{\Omega(t)} \nabla\varphi \cdot \nabla v d\Omega = \int_{\Gamma_p(t)} v\frac{i(\varphi)}{\sigma_c}ds, \quad \forall v \in V, \quad (3.17)$$

where $\Gamma_p(t)$ is the boundary of the pit at time t. Here V_h denotes a finite dimensional subspace of V spanned by a collection of finitely many basis functions (often associated with a mesh). We discretize the weak form (3.17) to find a solution in the discrete trial space. The discrete FEM solution is then found by finding $\varphi_h \in V_h \subset V$, such that

$$\int_{\Omega(t)} \nabla\varphi_h \cdot \nabla v_h d\Omega = \int_{\Gamma_p(t)} \frac{i(\varphi_h)}{\sigma_c} v_h ds, \quad \forall v_h \in V. \tag{3.18}$$

We can solve the discrete variational problem (3.18) in the following way. First, introduce $\{\phi_j\}_{j=1}^N$ as a basis for V_h and V. Let $\varphi_h \in V_h$ be a linear combination of the basis functions $\phi_j, j = 1, 2, \ldots, N$, with coefficients $\tilde{\varphi}_j$, given by

$$\varphi_h = \sum_{j=1}^{N} \tilde{\varphi}_j \phi_j. \tag{3.19}$$

Considering $v = \phi_k$, for $k = 1, 2, \ldots, N$, and using relation (3.19) gives

$$\sum_{j=1}^{N} \tilde{\varphi}_j \int_{\Omega} \nabla\phi_j . \nabla\phi_k d\Omega - \frac{1}{\sigma_c} \int_{\Gamma_p} i\Big(\sum_{j=1}^{N} \tilde{\varphi}_j \phi_j\Big) \phi_k ds = 0, \quad k = 1, 2, \ldots, N.$$

For each time, t, this is a system of non-linear equations which is solved using Newton's method.

3.3.2 The choice of the mesh density function

The appropriate specification of the mesh density tensor is crucial — it controls how the mesh automatically adapts to the changing solution features.

As mentioned, for problems with fixed domain boundaries, the Hessian based monitor function is an often used, general purpose, driver of the adaptive mesh. Here, however, we wish to ensure sufficient resolution of the pit geometry. To ensure a sufficient number of elements in the evolving pit and near the pit boundary, we use a modified version of MacKenzie's distance-based monitor function

$$\mathbb{M}_K(x, y) = \left(1 + \frac{\mu_1}{\sqrt{\mu_2^2 d^2 + 1}}\right) I, \tag{3.20}$$

where

$$d(x, y) = \min_{p} \|(x, y) - (x_p, y_p)\|,$$

and (x_p, y_p) denotes any point on the boundary of the pit, Γ_{pit}, cf. [4]. At any point $(x, y) \in \Omega$ the value of the monitor function involves the minimum distance, measured in the two-norm $\| \cdot \|$, from (x, y) to any point on the pit boundary. The reciprocal of $\mathbb{M}_K$ indicates that $\mathbb{M}_K$ will be largest in (x, y) regions where the distance to the pit boundary is the smallest, and hence the mesh spacing will be automatically smaller in these regions. The parameter μ_1 controls the minimum mesh spacing whereas μ_2 (and τ) will control the rate at which mesh clustering occurs during the integration of the MMPDE [33].

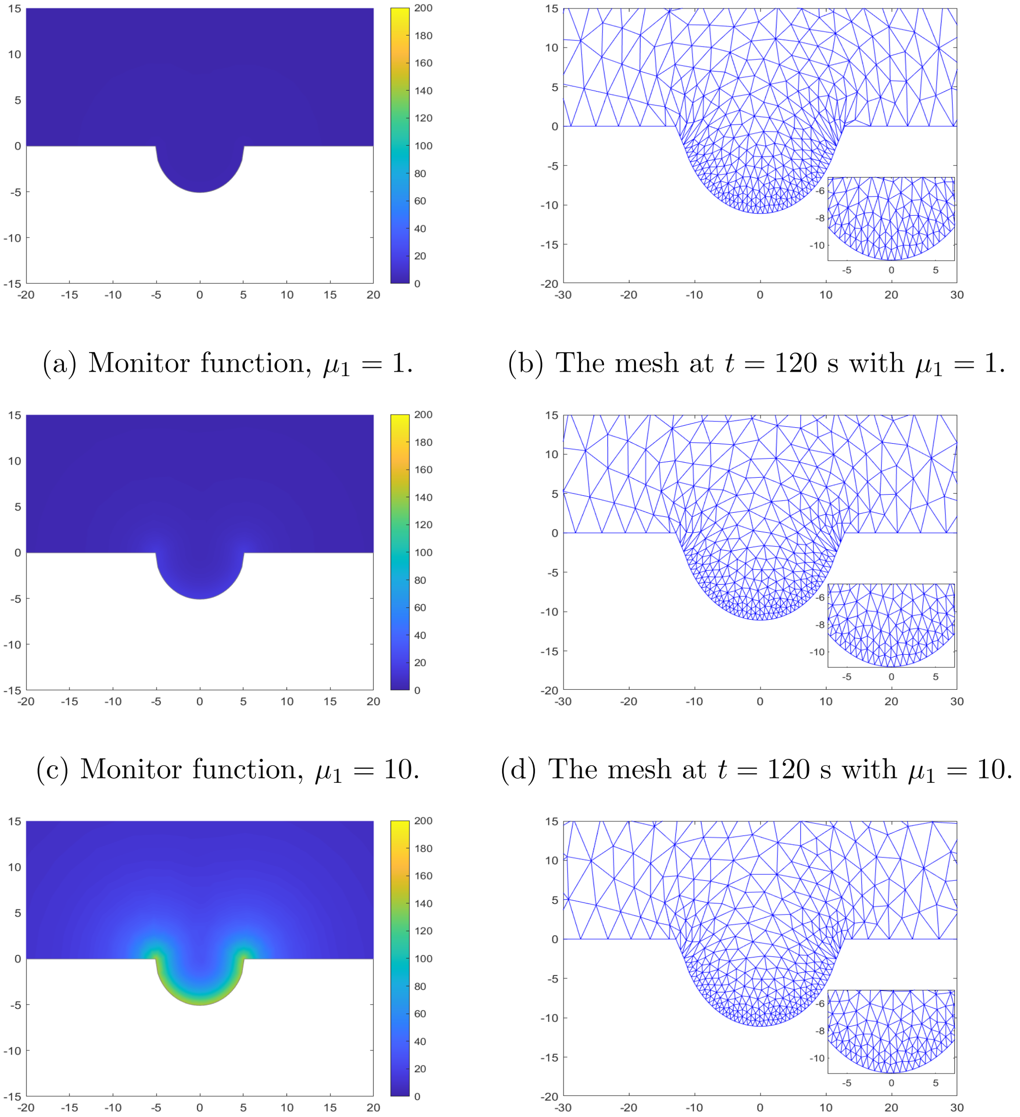
To understand the effect of the μ_1 and μ_2 parameters we consider a simple experiment, simulating the evolution of a single pit in a homogeneous material. A quality initial mesh, generated using the process outlined in Section 3.3.3 is used for this experiment.

We begin by fixing μ_2, at $\mu_2 = 1$, and consider the effect of increasing μ_1. The plots on the left of each row in Figure 3.3 show a representative mesh density function (computed at $t = 2s$), while the computed mesh after $t = 120s$ is shown on the right. The results show that increasing μ_1 leads to a monitor function which is (relatively) larger near the pit boundary and hence gives smaller grid spacings near the pit boundary. The value $\mu_1 = 100$ provides a balance between increased mesh density near the pit boundary and sufficient resolution throughout the rest of the computational domain.

Mackenzie [33] reports that increasing μ_2 reduces the spatial extent of node clustering near the pit boundary. To explore this we fix $\mu_1 = 100$ and vary μ_2, recording representative mesh density function values and the final mesh obtained for the propogation of a homogeneous pit is shown in Figure 3.4. The figure shows that increasing μ_2 to 20 is better able to keep the mesh focused on the feature of interest (the pit boundary in this case), consistent with the general findings in [33].

3.3.3 Initial mesh generation

The numerical simulations in this paper require an initial pit geometry and an initial spatial grid. With a prescription of the spatial domain many software platforms



(a) Monitor function, $\mu_1 = 1$.

(b) The mesh at $t = 120$ s with $\mu_1 = 1$.

(c) Monitor function, $\mu_1 = 10$.

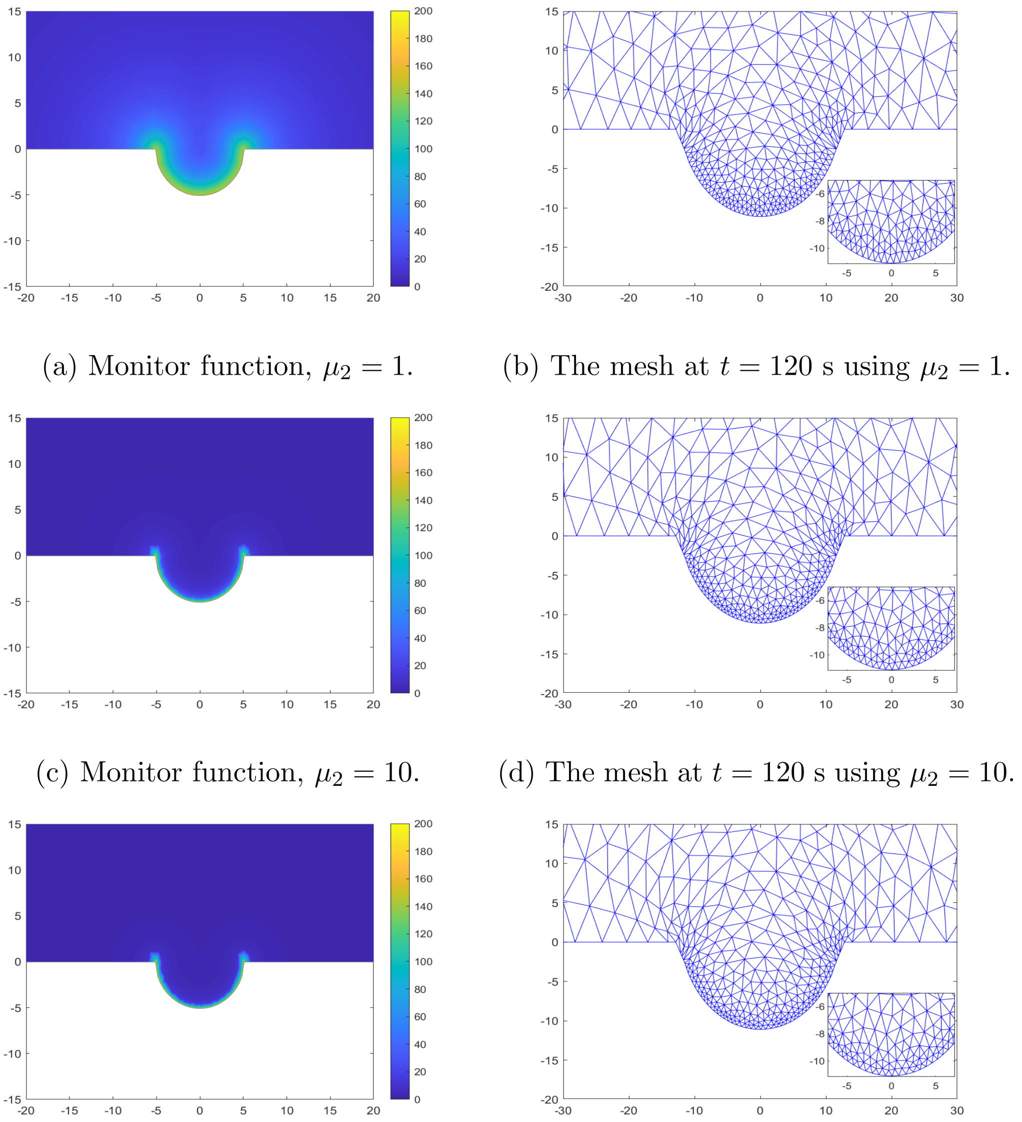
(d) The mesh at $t = 120$ s with $\mu_1 = 10$.

(e) Monitor function, $\mu_1 = 100$.

(f) The mesh at $t = 120$ s with $\mu_1 = 100$.

Figure 3.3: Effect of μ_1 on the mesh at $t = 120$ s for the simulation of a pit in a homogeneous material with the monitor function (3.20) and $\mu_2 = 1$.

provide tools known as mesh generators for this purpose, for example **initmesh** in Matlab or the **mesh node** in COMSOL. These tools require a description of the domain boundary and then generate a mesh subject to constraints on the mesh size (or number of nodes) and aspect ratios of the mesh elements.



(a) Monitor function, $\mu_2 = 1$.

(b) The mesh at $t = 120$ s using $\mu_2 = 1$.

(c) Monitor function, $\mu_2 = 10$.

(d) The mesh at $t = 120$ s using $\mu_2 = 10$.

(e) Monitor function, $\mu_2 = 20$.

(f) The mesh at $t = 120$ s using $\mu_2 = 20$.

Figure 3.4: Effect of μ_2 on the mesh at $t = 120$ s during the simulation of a pit in a homogeneous material with the monitor function (3.20) and using $\mu_1 = 100$.

In most cases the solutions found on the initial meshes generated by these approaches will not be optimal — for example, there is no guarantee that the error in the numerical solution will be minimized. There are alternatives for initial mesh generation that involve the MMPDE approach considered in this paper, and hence are consistent with the technique used for all subsequent time steps.

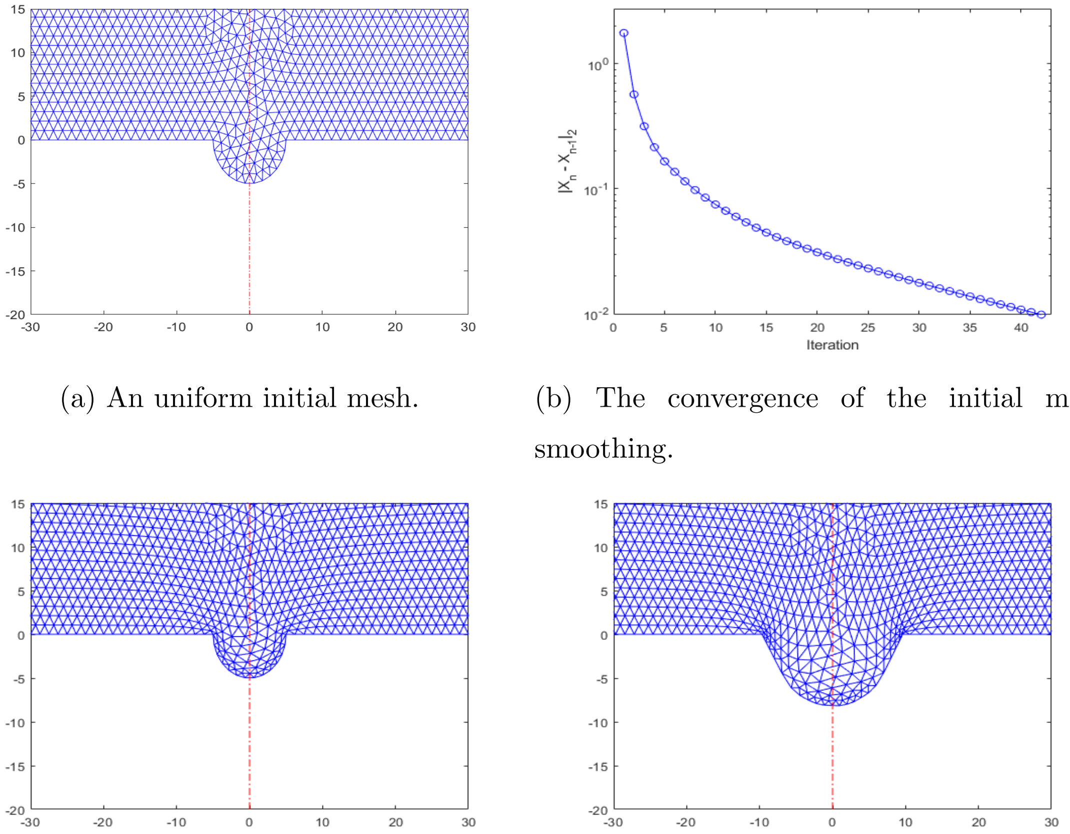
One technique for initial mesh tuning begins by using the simple mesh generators mentioned above to find a (nearly) uniform mesh. The physical problem defined by equations (3.3) and (3.4) is then solved on this mesh to give an initial potential. Using this initial potential and its associated mesh density function $\mathbb{M}_K$, the gradient flow equation (3.16) can be solved to a steady state (alternating its solution with physical solves). The result is a mesh which minimizes the discrete functional (3.15), equidistributing the initial potential over the initial computational domain. Should a more sophisticated non-uniform mesh generator be available, then an initial non–uniform mesh can be smoothed in the same manner. In practice, equation (3.16) may not solved to a steady state. Instead, equation (3.16) can be integrated for a specified number of time steps, or can be integrated until a specified difference between two meshes is found. We will call this mesh smoothing. The number of steps required to reach a suitable approximation of the steady state is a function of the physical solution, the number of mesh nodes, and the mesh density parameters. This idea of using the MMPDE to tune the initial mesh has the added benefit of giving a mesh which has the same properties as all subsequent meshes, while using the same code base as the rest of the simulation. We note that this process can also be used to provide small scale mesh smoothing during the solution of the moving boundary problem. This is particularly useful should the pit boundary movement be large or if the pit boundary movement induces a discontinuous change in the geometry (during pit merging for example).

In Figure 3.5 we show the evolution of the pit geometry in a homogeneous material. The moving mesh method is implemented with the monitor function (3.20) starting from the initial uniform mesh shown in Figure 3.5a. The non-uniform mesh resulting from the solution of the gradient flow equation, shown in Figure 3.5c, has successfully concentrated the mesh elements in the initial pit and near the initial pit boundary. The convergence to a steady state solution of the gradient flow equation is shown in Figure 3.5b. This resulting non-uniform mesh is now an appropriate initial mesh to use to evolve the pitting corrosion problem forward in time. It is important to stress that no hand-tuning of the initial mesh is necessary, it is generated automatically during the smoothing process based on the characteristics of the chosen mesh density function. The final mesh after $t = 60$ s is given in Figure 3.5d.

Presented in Figure 3.6a is the non-uniform initial mesh generated with the Matlab function **initmesh** where 45 nodes are located on the pit boundary. In order to optimize this mesh, smoothing steps were performed and the sum of the absolute differences in position of the nodes between subsequent smoothing iterations is displayed in Figure 3.6b. The motion of the nodes decreases with iteration number and after 17 iterations the absolute difference is down to 10^{-2}. The optimized mesh is shown in Figure 3.6c and significant differences in the locations of the nodes both inside and outside the pit are observed. Note that this smoothing operation only needs to be performed once since the results can be saved and used as the starting mesh in subsequent experiments. The computational mesh after the pit has evolved for 60 s is presented in Figure 3.6d and it is observed that node spacing within the pit remains very good.

The results in Figures 3.5 and 3.6 show that the MMPDE approach is robust with respect to the initial grid, continuously evolving the mesh according to changing domain and solution features. Even with a uniform initial mesh, the MMPDE approach does quite well, automatically recovering the requested increased mesh density near the pit boundary. We do notice, however, some additional stretching of the nodes in this case as compared to the simulation which starts from an improved non-uniform initial mesh. The stretching of the mesh can be reduced through the use of monitor functions designed to control the shape of the elements.

The relatively small scale mesh smoothing is particularly useful should the pit boundary movement be large or if the pit boundary movement induces a discontinuous change in the geometry (during pit merging for example). We will see this in Section 3.4.



(a) An uniform initial mesh.

(b) The convergence of the initial mesh smoothing.

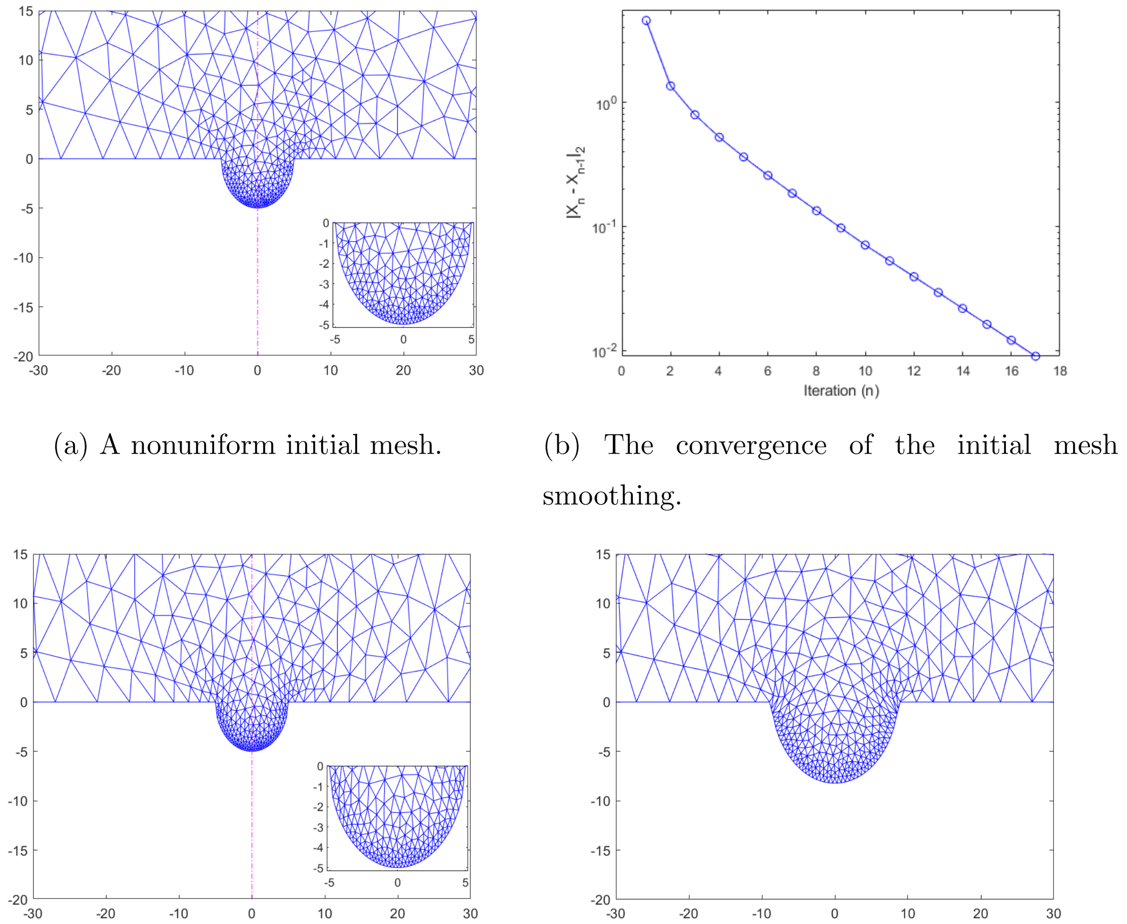
(c) The smoothed initial mesh.

(d) The mesh after 60 s.

Figure 3.5: (a) Uniform initial mesh, (b) convergence of the mesh smoothing process (c), initial mesh after mesh smoothing, and (d) the mesh after 60 s using the monitor function (3.20).

3.3.4 Effect of τ on the moving mesh

At the end of Section 2.3 we mentioned the MMPDE (relaxation) parameter τ and here we demonstrate the effect of τ on the moving mesh. Instead of forcing exact equidistribution each time t, we relax the condition and require equidistribution at time $t+\tau$. Hence, the smaller the size of τ the quicker the mesh will react to changing features in the solution. To demonstrate this effect, we start from a uniform initial grid and show the resulting meshes after $t = 60$ s using three values of τ, as shown in Figure 3.7. We observe a greater concentration of nodes near the pit boundary for



(a) A nonuniform initial mesh.

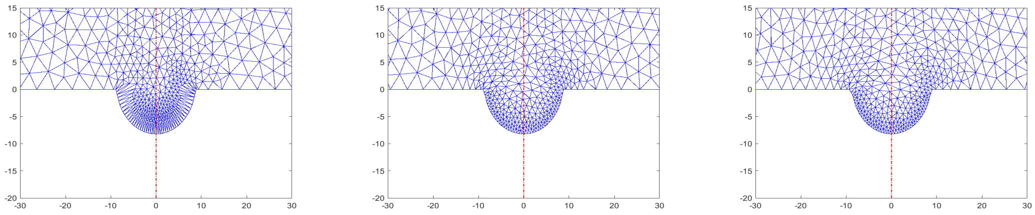
(b) The convergence of the initial mesh smoothing.

(c) The smoothed initial mesh.

(d) The mesh after $t = 60$ s.

Figure 3.6: (a) A nonuniform initial mesh, (b) the effect of mesh smoothing on the positions of the nodes, (c) the mesh after smoothing, and (d) the mesh a $t = 60$ s using the monitor function (3.20).

smaller values of τ. Larger values of τ lead to stretched elements; the mesh is not able to keep up with the changing computational domain. This relaxation does come at a cost, however, as smaller values of τ require more time steps for the integration of the MMPDE. In practice, one should select τ in tandem with mesh density parameters, choosing the largest value of τ which allows a balance of computational cost and mesh quality.



(a) Using $\tau = 10^{-2}$.

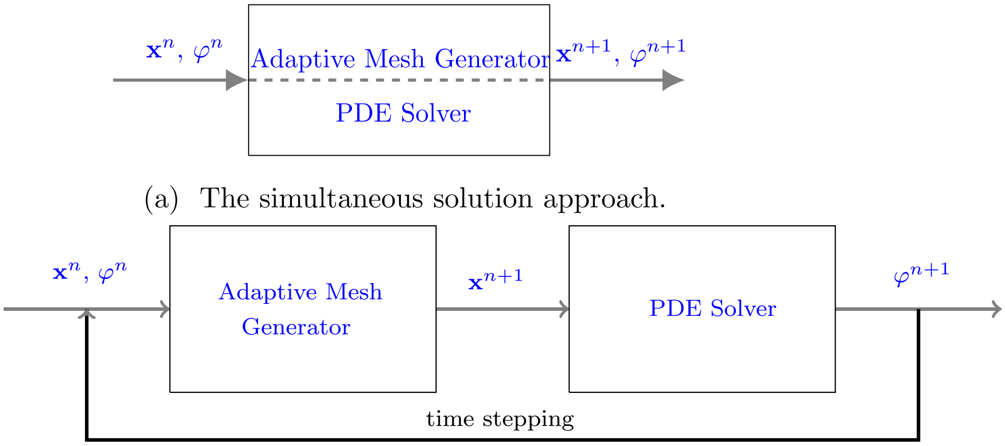
(b) Using $\tau = 10^{-4}$.

(c) Using $\tau = 10^{-6}$.

Figure 3.7: Effect of τ on the mesh after 120 s with the monitor function (3.20) using $\mu_1 = 100$ and $\mu_2 = 1$.

3.3.5 Alternating mesh and physical PDE iteration

There are two approaches that can be used to solve the coupled physical PDE and mesh equation: a simultaneous or an alternating approach. In a simultaneous approach, the discrete physical PDE and the discrete mesh equation provide a fully coupled system for both the mesh and solution unknowns, as shown in Figure 3.8a. The disadvantage of this approach is the highly nonlinear coupling between the physical solution and the mesh, resulting in a potentially difficult, large discrete system.



(a) The simultaneous solution approach.

(b) The alternating solution approach.

Figure 3.8: The (a) simultaneous and (b) alternating approaches to solve the coupled corrosion model and mesh PDE.

On the other hand, the alternating solution approach generates the mesh $\mathbf{x}^{n+1}$ at a new time step using the physical solution φ^n and the mesh $\mathbf{x}^n$ at the current time. Then the solution φ^{n+1} at the new time level is computed, as shown in Figure 3.8b. In this approach, there may be a lag between the solution and the mesh. Generally, this does not create any difficulties if the time step is reasonably small. The main advantages of the alternating approach are: (i) the mesh generation code is not directly coupled to the physical PDE solve thereby increasing flexibility and reusability of code, (ii) the mesh PDE and physical PDE solvers can be developed and optimized in a modular way, and hence (iii) the individual mesh and physical PDE solvers are more efficient. MMPDElab uses this alternating approach.

3.3.6 Solution of the moving boundary value problem

The flowchart in Figure 3.9 outlines the implementation of our computational pitting corrosion model using MMPDElab framework. The first two steps are the same as the alternating step approach given in Figure 3.8b. The pit boundary is then moved based on the new positions of the adaptive mesh, followed by the movement of the corners of the pit. These last steps are detailed in the following section.

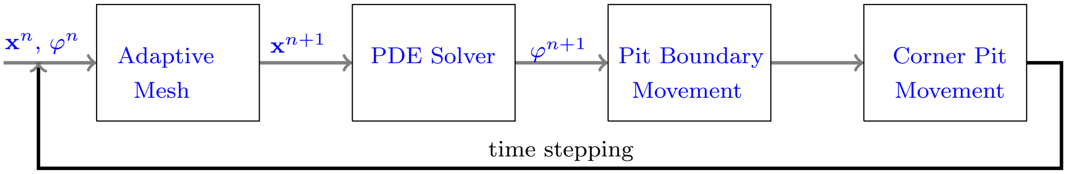


Figure 3.9: Flow chart for the physical PDE solve, the mesh PDE solve, and the pit boundary movement.

3.3.7 Details of the pit boundary movement

To get the new position of the pit we have to specify a direction and magnitude of movement for each node on the boundary of the pit and the appropriate movement for each corner of the pit. A pit corner is a vertex which is part of the pit boundary and has a y–coordinate of zero.

As shown in Figure 3.10, a face normal is the outward pointing vector perpendicular to an edge or segment joining vertices. Taking the average of two face normals on adjacent edges gives us the vertex normal for the vertex between those edges. The vertex normals give the direction of movement for the pit boundary.

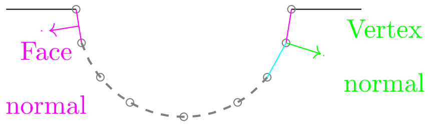


Figure 3.10: Definitions of face and vertex normals.

As mentioned in Section 2.1, the magnitude of the normal velocity of each vertex on the boundary of the pit is given by (3.6), which we may write as

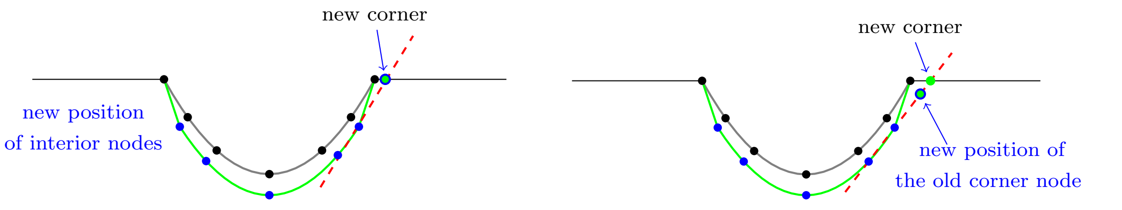
$$V_n = \frac{1}{c_{\text{solid}}} \cdot A_{\text{diss}} \cdot e^{\left(\frac{zF(V_{\text{corr}}+\alpha(V_{\text{app}}-V_{\text{corr}}-\varphi))}{RT}\right)}. \tag{3.21}$$

Once the vertices on the pit boundary are moved, the location of the corner nodes for the pit are updated using the following procedure. A linear extrapolation of the edge joining the two vertices that are closest to the corner and lie on the pit boundary is computed. The new corner location is given by the intersection of this line and $y = 0$, as necessary.

There are two situations which may arise as shown in Figure 3.11. If the new corner is close to the old corner (Figure 3.11a), then no further changes are required. If the new corner is not close to the old corner (Figure 3.11b) then the old corner is moved into the pit using the same extrapolation line. The idea here is to support large movements of the pit boundary by moving grid points along $y = 0$ into the pit boundary.

3.3.8 Merging pits

As multiple pits evolve, two pits may merge and form a single, larger pit. We visualize the pit merging procedure in Figure 3.13. The merging process is initiated (Figure



(a) New corner remains on $y = 0$.

(b) Corner node movement onto the pit boundary.

Figure 3.11: Updating the corner position: (a) the corner is moved to its new location along $y = 0$ or (b) the (old) corner is moved onto the boundary of the pit.

3.13a) when there is a single edge between two pits that is less than the user prescribed tolerance. The left and right endpoints of that edge are tagged with red and black, respectively. In Figure 3.13b these two points merge to a single point. In order to avoid changing the number of mesh nodes and mesh topology, either the black or the red node in Figure 3.13a has to move into the pit boundary either on the left or the right side of the apex. For example, the black and red nodes can merge, creating a new red node (see Figure 3.13b), and the black node will move half-way between the red and green nodes. Alternately, the black and red node merge, creating a new black node and the red node moves half-way between the red node and green node (not shown). We choose the vertex corresponding to the larger angle in the element whose bottom edge is the single edge between the pits, see Figure 3.12. Once the merge has occurred, a mesh smoothing procedure is used to obtain the mesh shown in Figure 3.13c. We can see that the mesh smoothing has evened out the size and shape of the elements to the right of the red apex node.

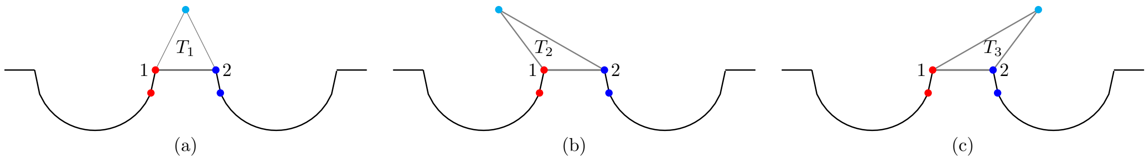
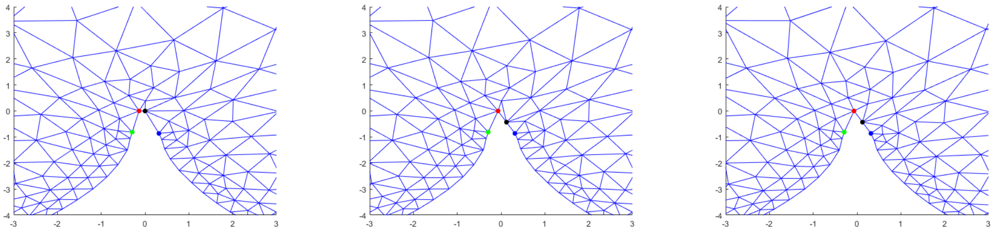


Figure 3.12: Three possible element orientations between the pits at the time of a merge



(a) A pit merge is initiated. (b) A merge with no mesh topology change. (c) Post merge mesh smoothing.

Figure 3.13: The pit merging process.

For each time step after the merge the location of the apex is obtained as the intersection point of the linear extrapolations of the second last edges to the left and right of the apex.

3.4 Numerical results

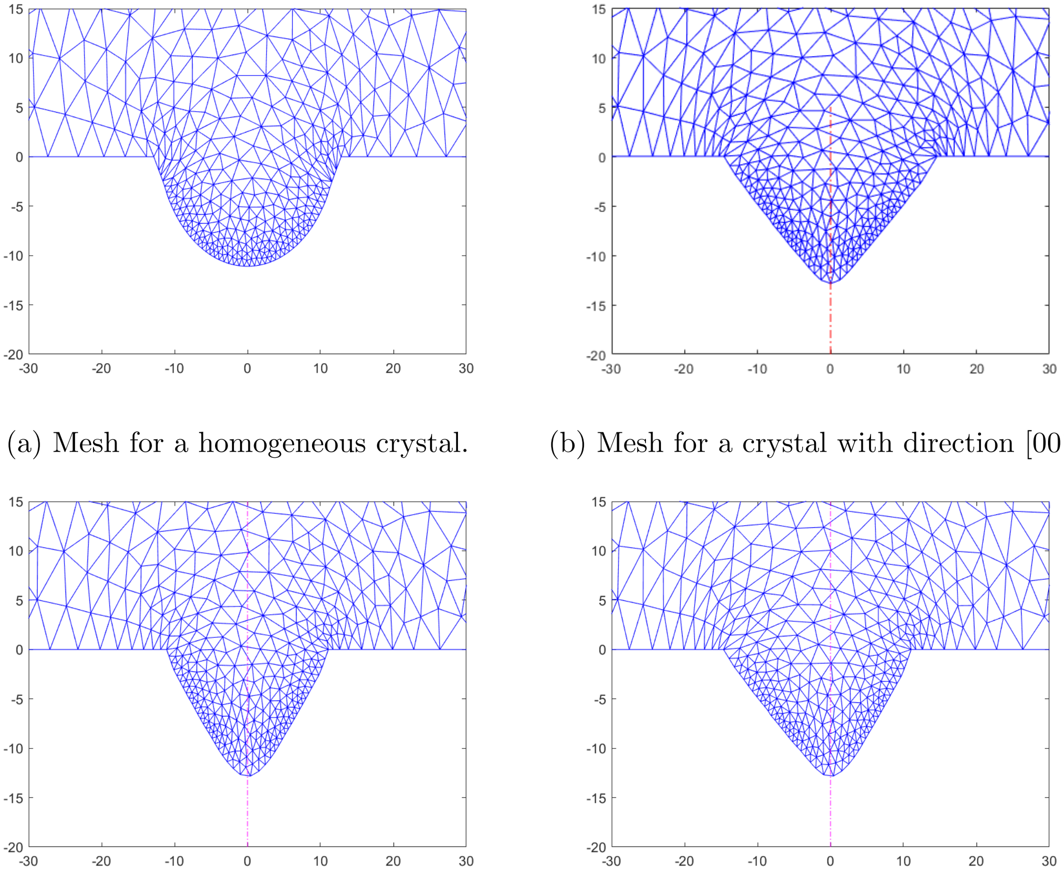
Based on the simple experiments in Section 3, throughout this numerical results section a constant value of $\tau = 10^{-5}$ is used in the MMPDE, and the constants $\mu_1 = 100$ and $\mu_2 = 1$ are used as default values for the mesh density function (3.20). The initial number of mesh points inside the pit is set to 61.

3.4.1 Single pit simulations

We begin by using our computational framework to compare the evolution of a single pit in three cases: a homogeneous solid material (without a crystal direction), a solid material with a specified crystal direction, and a solid material with a discontinuity in the crystal direction. All simulations use an initial mesh constructed by smoothing the non-uniform mesh generated using **initmesh** as discussed in Section 3.3.

Figures 3.14a 3.14b, 3.14c and 3.14d show the final pit geometry and final meshes for a homogeneous material, a material with crystal direction [001], a material with crystal direction [101], and a material with a discontinuity in crystal directions, [001] to the left of $x = 0$ and [101] to the right of $x = 0$.

It is observed in Figure 3.14a that the final shape of the homogeneous pit is the same as the initial pit since the chosen V_{corr} is a constant value of -0.24 V, that is, the same in every direction within the pit. Hence, from equation (3.21) the normal velocity at all locations within the pit will be equal. The situation is not the same for pits with a crystalline structure since V_{corr} will vary with crystal direction. For example, for crystal directions of the forms $\langle 001 \rangle$, $\langle 011 \rangle$ and $\langle 111 \rangle$, V_{corr} will have values of -0.2297 V, -0.2455 V, and -0.2525 V, respectively. Again from equation (3.21), we see that the normal velocity is greater for lower magnitude V_{corr} values; that is, $V_n\,(111) < V_n\,(011) < V_n\,(001)$. The effect of this V_{corr} dependent velocity is displayed in Figure 3.14b where we observe that the sides of the pit have become straight and angled 90 degrees with one another (when axes are equally scaled). This behaviour is expected. As shown in Figure 3.2, for a crystal oriented with a zone axis along [001], $\langle 001 \rangle$ directions are located along the horizontal and vertical axes and $\langle 011 \rangle$ directions midway in between. Thus, we expect that the [100] and $[0\bar{1}0]$ directions will move faster than the $[1\bar{1}0]$ direction. As the faster locations on the pit boundary move, their orientation will change and eventually become the same direction as the slowest moving axis, in this case $[1\bar{1}0]$. For all future times, the sides of the pit will move outward perpendicular to these two lines while maintaining the same angular relationship. We observe the same effect in Figure 3.14c where the crystal has been oriented along a zone axis of [101]. In this case, the slowest moving directions are along $\langle 111 \rangle$ and the angle between the $[11\bar{1}]$ and $[\bar{1}11]$ planes agrees with the expected value of 70.5 degrees. Figure 3.14d displays the final pit shape where there is a discontinuity in the crystal directions and the left and right sides of the crystal were oriented along zone axes of [001] and [101], respectively. The left and right sides of the pit are straight lines moving along $[\bar{1}\bar{1}0]$ and $[\bar{1}11]$ directions, respectively.



(a) Mesh for a homogeneous crystal.

(b) Mesh for a crystal with direction [001].

(c) Mesh for a crystal with direction [101].

(d) Mesh for a crystal with two directions [001] and [101].

Figure 3.14: Pit configurations and meshes at $t = 120$ s for a) a homogeneous material, b) a single crystal oriented with a zone axis along [001], c) a single crystal oriented with a zone axis along [101], and d) a crystal with an interface at $x = 0$; the crystal directions to the left and right of $x = 0$ are [001] and [101], respectively.

3.4.2 Multiple pit simulations

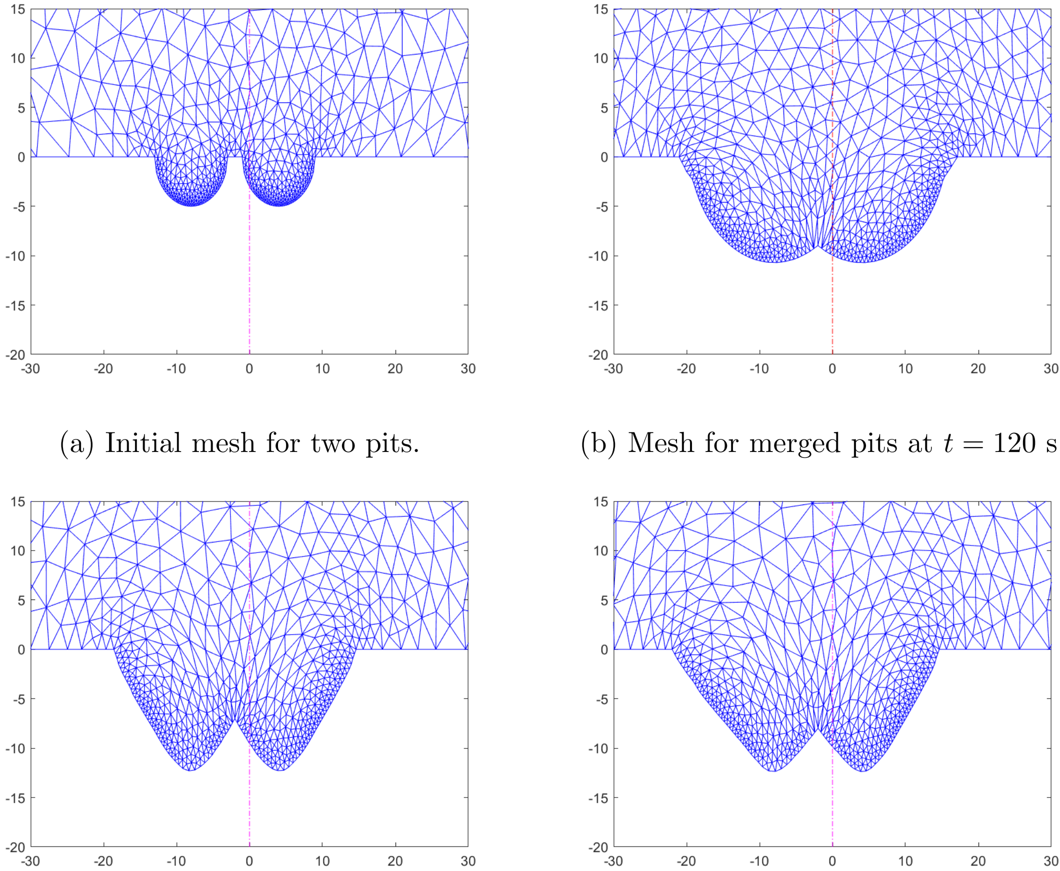
As mentioned previously, if multiple pits exist in a material and the pits grow large enough, there is the potential that the pits will merge during the simulation. The imminent merge needs to be detected, the boundaries of each of the previously isolated pits need to be updated in a way which avoids boundary crossing, and the mesh around merge location needs to be adjusted smoothly and without topology changes. Once complete the merged pit is then treated as one larger pit, and the evolution continues. See Section 3.3.8 for details.

To demonstrate the robustness of our adaptive simulation framework for the evolution of multiple pits we start with two pits relatively close together in a homogeneous material, as shown in plot Figure 3.15a. A quality initial mesh which concentrates nodes near the boundary of both pits is generated as discussed in Section 3.3.3. As the corrosion continues the pits grow, and hence grow closer together. The pits then merge and continue to evolve as shown in Figure 3.15b. Figure 3.15c provides the result of a similar simulation with a material oriented in the [101] crystal direction. The crystal direction clearly affects the geometry of the merged pit. Figure 3.15d shows the resulting pit geometry and associated mesh for merged pits in a material with two crystal directions, where the discontinuity in crystal direction is located at $x = 0$.

We now more closely study the pit depth and width as a function of time for a single homogeneous pit, a single crystal oriented with a zone axis along [001], and two crystals with an interface at $x = 0$ where the left and right crystals are oriented along [001] and [101] zone axes, respectively. Recall, the initial and final pit configurations for these three situations are displayed in Figures 3.14a, 3.14b and 3.14d, respectively. Plotting pit depth and width as a function of time leads to nonlinear curves as shown in Figures 3.16a and 3.16b. It has been common practice to fit corrosion loss curves using a power-law equation and for the initial stages of corrosion it seems to work well, see [36]. Since the loss of material due to corrosion is a function of the dimensions of the pit, it is expected that the same power-law behaviour should hold for our pit depth and width data. The model we use is

$$\text{depth(t)} \ \ \text{(or)} \ \ \text{width(t)} = at^b + c,$$

where a, b and c are fitting parameters and the initial pit is defined when $t = 1$. The curve fits were excellent and the fitting parameters found are presented in Table 2. The initial pit width and depth were 10 microns and 5 microns, respectively, and



(a) Initial mesh for two pits.

(b) Mesh for merged pits at $t = 120$ s.

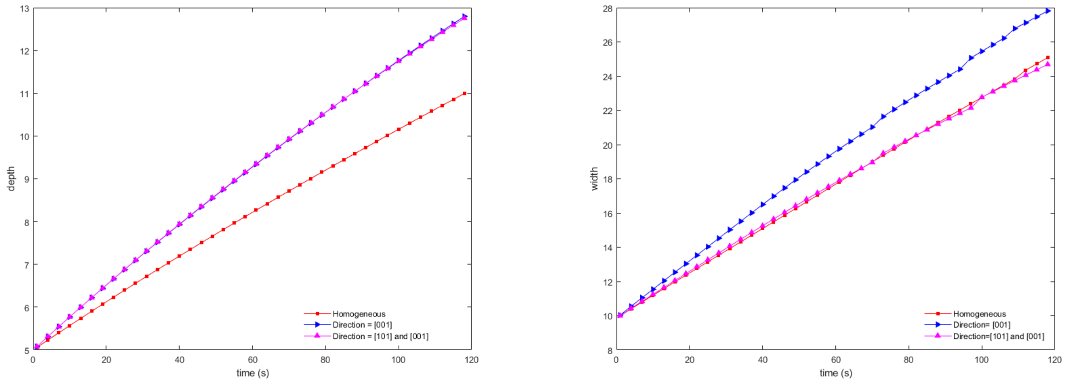
(c) Mesh for merged pits for a material with a single crystal direction [101] at $t = 120$ s.

(d) Mesh for merged pits for a material with two crystal directions, [001] if $x < 0$ and [101] if $x > 0$, at $t = 120$ s.

Figure 3.15: Pit evolution and adaptive mesh generation for merging multiple pits for three material configurations.

these values are close to the value of $a + c$; the initial dimension of the pit predicted from the fitting procedure. It is reassuring to note that the modelled pitting corrosion behaviour follows an expression used to fit experimental corrosion losses.

Figure 3.17 displays the pit depths and pit widths for various initial numbers of mesh points using both the moving mesh approach and the re-meshing technique at each time step. For a lower number of initial mesh points (i.e., INPPit = 31) on the pit, do not match the pit depth and width obtained by using the remeshing technique



(a) Pit-depth over time for homogeneous and non-homogeneous crystals.

(b) Pit-width over time for homogeneous and non-homogeneous crystals.

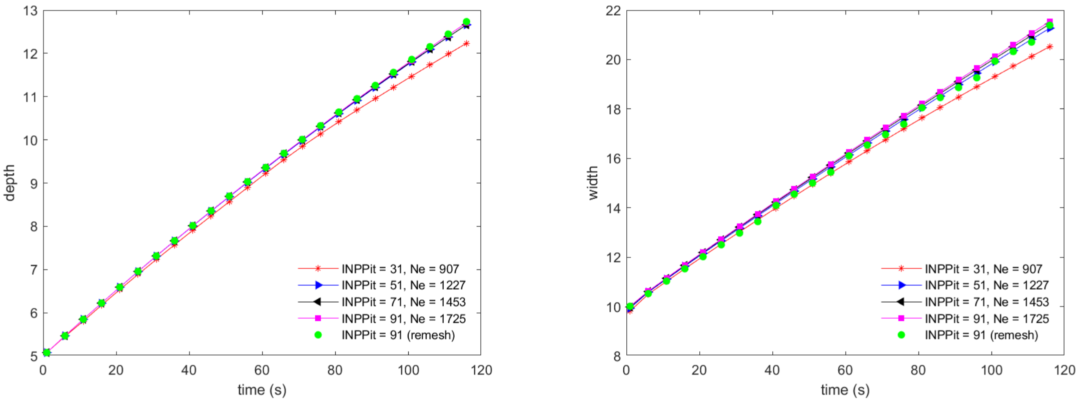
Figure 3.16: Pit-depths and widths for homogeneous and non-homogeneous crystals.

Case	**Width**			**Depth**		
	a	b	c	a	b	c
Homogeneous	0.142(2)	0.980(3)	9.83(2)	0.076(1)	0.917(3)	4.95(1)
[001]	0.243(8)	0.907(6)	9.62(5)	0.116(3)	0.886(4)	4.89(2)
[001]/[101]	0.181(5)	0.927(6)	9.74(4)	0.121(3)	0.877(6)	4.89(2)

Table 3.2: Power-law model fitting parameters for the 6 curves presented in Figure 3.16. The numbers in brackets represent uncertainty in the last significant digit.

at every time step as time increases. However, with more than 50 initial mesh points on the pit, the pit depths and pit widths for both approaches converge to the same solution. Thus, we can conclude that the adaptive moving mesh approach provides a reasonable solution accuracy by delivering precise pit depth and width measurements.

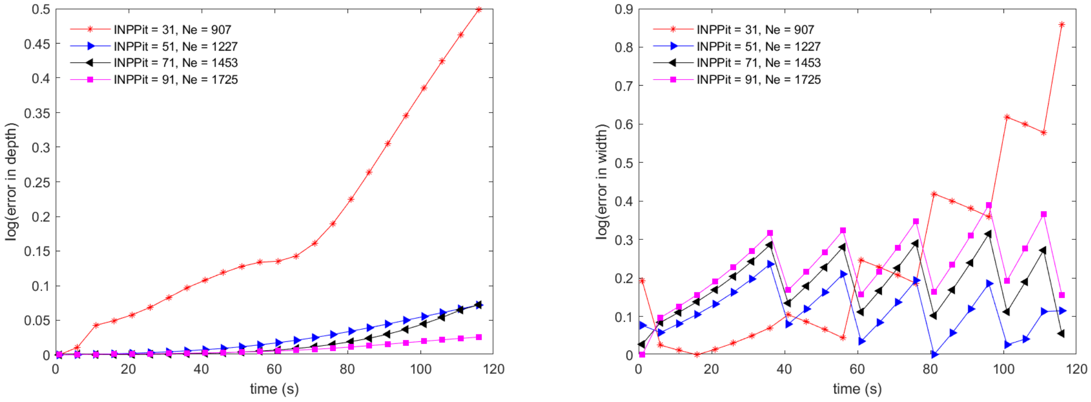
Figure 3.17 shows the numerical error in pit depths and widths when comparing the remeshing technique with the adaptive moving mesh approach for various initial numbers of mesh points on the pit. The absolute error is calculated by comparing the remeshing solution with the moving mesh solution at every step. For a lower number



(a) Pit-depth over time.

(b) Pit-width over time.

Figure 3.17: Pit-depths and widths for homogeneous case for various initial number of mesh point on the pit.



(a) Error in pit depth.

(b) Error in pit depth.

Figure 3.18: The numerical error for pit depths and widths for various initial numbers of mesh points on the pit.

of initial mesh points (i.e., INPPit = 31) on the pit, the error increases over time. However, with more than 50 initial mesh points on the pit, the error becomes bounded. Certainly, the adaptive moving mesh is able to provide a reasonable solution, as shown in Figure 3.17. These numerical results ensure that the adaptive moving mesh is a suitable choice for solving the moving boundary problem related to pitting corrosion.

3.5 Conclusion

We have presented a robust, fully automatic, moving mesh solution framework for pitting corrosion. The moving mesh approach continuously and smoothly evolves a fixed mesh topology according to changing pit geometry. Single and multiple pits are considered, as are materials with different crystal direction(s). A procedure is presented which allows pits to merge without a change in mesh topology, allowing computation to proceed without restarting the computation.

The simulation of large pit growth or the initiation of many pits would likely benefit from an *hr*–refinement strategy (which both redistributes nodes as we have presented here but also allows periodic changes to the number of mesh nodes) coupled with a domain decomposition approach to allow the problem to be spatially partitioned and the computation distributed to harness additional processors. This will be the subject of future work. Current work includes extending the computational framework to allow for more heterogenous materials, with corrosive resistant "pockets" or holes or voids.

3.6 Data availability

The raw or processed data required to reproduce these figures and findings cannot be shared at this time due to technical or time limitations, but are available from the authors upon request.

3.7 Acknowledgements

The work of Ronald D. Haynes and Abu N. Sarker (PhD student) has been supported by a NSERC discovery grant (Canada).

Bibliography

[1] David R Adams and Lars I Hedberg. *Function Spaces and Potential Theory*, volume 314. Springer Science & Business Media, 1999.

[2] Allen J Bard, Larry R Faulkner, Johna Leddy, and Cynthia G Zoski. *Electrochemical Methods: Fundamentals and Applications*, volume 2. Wiley New York, 1980.

[3] Allen J Bard, Larry R Faulkner, and Henry S White. *Electrochemical Methods: Fundamentals and Applications*. John Wiley & Sons, 2022.

[4] G Beckett, John A Mackenzie, and ML Robertson. A moving mesh finite element method for the solution of two-dimensional Stefan problems. *Journal of Computational Physics*, 168(2):500–518, 2001.

[5] Monte B Boisen and Gerald V Gibbs. *Mathematical Crystallography*, volume 15. Walter de Gruyter GmbH & Co KG, 2018.

[6] Jeremiah U Brackbill and Jeff S Saltzman. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics*, 46(3):342–368, 1982.

[7] Weiming Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM Journal on Numerical Analysis*, 43(1):19–40, 2005.

[8] Weiming Cao, Weizhang Huang, and Robert D Russell. A study of monitor functions for two-dimensional adaptive mesh generation. *SIAM Journal on Scientific Computing*, 20(6):1978–1994, 1999.

[9] François Cattant, Didier Crusset, and Damien Féron. Corrosion issues in nuclear industry today. *Materials Today*, 11(10):32–37, 2008.

[10] Dennj De Meo and Erkan Oterkus. Finite element implementation of a peridynamic pitting corrosion damage model. *Ocean Engineering*, 135:76–83, 2017.

[11] Virginia G DeGiorgi, Nithyanand Kota, Alexis C Lewis, and Siddiq M Qidwai. Numerical modeling of pit growth in microstructure. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55850, pages 1–7. American Society of Mechanical Engineers, 2013.

[12] Jean Donea, Antonio Huerta, J-Ph Ponthot, and Antonio Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. *Encyclopedia of Computational Mechanics*, 2004.

[13] Ravindra Duddu. Numerical modeling of corrosion pit propagation using the combined extended finite element and level set method. *Computational Mechanics*, 54(3):613–627, 2014.

[14] CW Hirt, AA Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.

[15] Cyrill W Hirt, Anthony A Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[16] Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *Journal of Computational Physics*, 174(2):903–924, 2001.

[17] Weizhang Huang. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics*, 204(2):633–665, 2005.

[18] Weizhang Huang. An introduction to MMPDElab. *ArXiv Preprint ArXiv:1904.05535*, 2019.

[19] Weizhang Huang, Yuhe Ren, and Robert D Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM Journal on Numerical Analysis*, 31(3):709–730, 1994.

[20] Weizhang Huang and Robert D Russell. A high dimensional moving mesh strategy. *Applied Numerical Mathematics*, 26(1-2):63–76, 1998.

[21] Weizhang Huang and Robert D Russell. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM Journal on Scientific Computing*, 20(3):998–1015, 1998.

[22] Weizhang Huang and Robert D Russell. *Adaptive Moving Mesh Methods*, volume 174. Springer Science & Business Media, 2010.

[23] Olivier-Pierre Jacquotte. A mechanical model for a new grid generation method in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 66(3):323–338, 1988.

[24] Olivier-Pierre Jacquotte. Generation, optimization and adaptation of multiblock grids around complex configurations in computational fluid dynamics. *International Journal for Numerical Methods in Engineering*, 34(2):443–454, 1992.

[25] Olivier-Pierre Jacquotte and Grégory Coussement. Structured mesh adaption: space accuracy and interpolation methods. *Computer Methods in Applied Mechanics and Engineering*, 101(1-3):397–432, 1992.

[26] Siavash Jafarzadeh, Ziguang Chen, and Florin Bobaru. Computational modeling of pitting corrosion. *Corrosion Reviews*, 37(5):419–439, 2019.

[27] Patrick M Knupp. Mesh generation using vector fields. *Journal of Computational Physics*, 119(1):142–148, 1995.

[28] Patrick M Knupp. Jacobian-weighted elliptic grid generation. *SIAM Journal on Scientific Computing*, 17(6):1475–1490, 1996.

[29] Patrick M Knupp, Len G Margolin, and Mikhail Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176(1):93–128, 2002.

[30] Nithyanand Kota, Siddiq M Qidwai, Alexis C Lewis, and Virginia G DeGiorgi. Microstructure-based numerical modeling of pitting corrosion in 316 stainless steel. *ECS Transactions*, 50(31):155–164, 2013.

[31] D Krouse, N Laycock, and C Padovani. Modelling pitting corrosion of stainless steel in atmospheric exposures to chloride containing environments. *Corrosion Engineering, Science and Technology*, 49(6):521–528, 2014.

[32] Nicholas J Laycock, Donal P Krouse, Shaun C Hendy, and David E Williams. Computer simulation of pitting corrosion of stainless steels. *The Electrochemical Society Interface*, 23(4):65–71, 2014.

[33] JA Mackenzie and ML Robertson. The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method. *Journal of Computational Physics*, 161(2):537–557, 2000.

[34] Abdel Salam H Makhlouf, Victor Herrera, and Edgar Muñoz. Corrosion and protection of the metallic structures in the petroleum industry due to corrosion and the techniques for protection. In *Handbook of Materials Failure Analysis*, pages 107–122. Elsevier, 2018.

[35] Len G Margolin. Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):198–202, 1997.

[36] Robert E Melchers. Predicting long-term corrosion of metal alloys in physical infrastructure. *npj Materials Degradation*, 3(1):1–7, Jan 2019.

[37] Fabio Nobile and Luca Formaggia. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-West Journal of Numerical Mathematics*, 7:105–132, 1999.

[38] Yuhe Ren and Robert D Russell. Moving mesh techniques based upon equidistribution, and their stability. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1265–1286, 1992.

[39] Pierre R Roberge. *Corrosion Engineering.* McGraw-Hill Education, 2008.

[40] M. Robertson and K. Raffel. Imaging crystals. *Microscopical Society of Canada Bulletin*, 36:13–18, 08 2008.

[41] Stefan Scheiner and Christian Hellmich. Stable pitting corrosion of stainless steel as diffusion-controlled dissolution process with a sharp moving electrode boundary. *Corrosion Science*, 49(2):319–346, 2007.

[42] Stefan Scheiner and Christian Hellmich. Finite volume model for diffusion-and activation-controlled pitting corrosion of stainless steel. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2898–2910, 2009.

[43] SM Sharland. A review of the theoretical modelling of crevice and pitting corrosion. *Corrosion Science*, 27(3):289–323, 1987.

[44] SM Sharland. A mathematical model of crevice and pitting corrosion—II. the mathematical solution. *Corrosion Science*, 28(6):621–630, 1988.

[45] SM Sharland, CP Jackson, and AJ Diver. A finite-element model of the propagation of corrosion crevices and pits. *Corrosion Science*, 29(9):1149–1166, 1989.

[46] SM Sharland and PW Tasker. A mathematical model of crevice and pitting corrosion-I. The physical model. *Corrosion Science*, 28(6):603–620, 1988.

[47] Laura B Simon, Mohammad Khobaib, Theodore E Matikas, CS Jeffcoate, and MS Donley. Influence of pitting corrosion on structural integrity of aluminum alloys. In *Nondestructive Evaluation of Aging Materials and Composites III*, volume 3585, pages 40–47. International Society for Optics and Photonics, 1999.

[48] Joe F Thompson, Frank C Thames, and C Wayne Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, 15(3):299–319, 1974.

[49] A Turnbull, DA Horner, and BJ Connolly. Challenges in modelling the evolution of stress corrosion cracks from pits. *Engineering Fracture Mechanics*, 76(5):633–640, 2009.

[50] John C Walton. Mathematical modeling of mass transport and chemical reaction in crevice and pitting corrosion. *Corrosion Science*, 30(8-9):915–928, 1990.

[51] Kai Wang, Chenpei Li, Yanhui Li, Jinling Lu, Yueshe Wang, and Xingqi Luo. Multi-physics coupling analysis on the time-dependent localized corrosion behavior of carbon steel in CO_2-H_2O environment. *Journal of The Electrochemical Society*, 167(1):013505–013505, 2019.

[52] BV Wells, Michael J Baines, and Paul Glaister. Generation of arbitrary Lagrangian–Eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations. *International Journal for Numerical Methods in Fluids*, 47(10-11):1375–1381, 2005.

[53] Alan M Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.

[54] Alan M Winslow. Adaptive-mesh zoning by the equipotential method. Technical report, Lawrence Livermore National Lab., CA (USA), 1981.

[55] Chaoyang Xie, Fayuan Wei, Pingfeng Wang, and Hongzhong Huang. Modeling of corrosion pit growth for prognostics and health management. In *Prognostics and Health Management (PHM), 2015 IEEE Conference on*, pages 1–7. IEEE, 2015.

[56] Xingliang Yu, Fangyi Wan, and Yingnan Guo. Micromechanics modeling of skin panel with pitting corrosion for aircraft structural health monitoring. In *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE, 2016.

Chapter 4

A moving mesh simulation for pitting corrosion of heterogeneous materials

Abstract

[1] The design and implementation of an adaptive moving mesh method is provided for the simulation of pitting corrosion for materials with heterogeneous inclusions. The adaptive mesh is generated automatically by solving a mesh PDE coupled to a nonlinear potential problem. The moving mesh approach is able to smoothly tackle the changing pit geometry associated with materials with inclusions of varying crystallography, with corrosion-resistant inclusions, and material voids. This project presented a robust, fully automatic, moving mesh solution framework for pitting corrosion in materials with heterogeneous inclusions.

Keywords: Pitting Corrosion, Adaptive Moving Mesh, MMPDE, FEM, Crystallography, Heterogeneous Materials, Corrosion-resistant Inclusions.

Declaration of Competing Interests: We have no competing interests to declare.

[1] This work is under revision as "*A moving mesh simulation for pitting corrosion of heterogeneous materials*", by A. N. Sarker, R. D. Haynes, and M. D. Robertson, to the Journal of Computers & Mathematics with Applications, 2023.

4.1 Introduction

Pitting corrosion is a particularly insidious, localized form of corrosion which generates a small pit, cavity or hole in the metal. Detecting pitting corrosion is difficult, and hence can impact the structural integrity of metal before it is observed [49, 58]. The shape and size of the pits depend on many factors: the components of the metal, the orientation of metal's surface, and the local chemical environment [46].

Researchers have studied pitting initiation behavior of different metals including stainless steel [12, 34, 42], beryllium [12], pure aluminum and aluminum alloys [27, 31, 45, 51, 57]. Davis [7] has shown the behavior of metastable pitting corrosion of aluminum single crystals.

Determining the pitting behavior experimentally is time consuming, expensive and physically difficult or impossible in many situations. Hence, numerical simulations have been used to efficiently study pitting corrosion under a wide range of conditions.

Recently, [8, 9, 11, 32, 33, 35, 43, 44, 44, 47, 48, 52, 53, 54] have provided FEM or finite volume solution approaches for partial differential equation (PDE) based models for pitting corrosion. An extensive overview [25] discusses models based on the anodic reaction at the corrosion front and the transportation of ions in the pits of the electrolyte domain. In many of these previous studies the commerical software COMSOL® is used to solve the PDE in the electrolyte domain.

Most of the previously mentioned FEM approaches relied on a complete remeshing of the domain at every time step provided by the arbitrary Lagrangian-Eulerian (ALE) approach. Here we provide an alternative approach which uses an adaptive moving mesh method, automatically and continuously varying the size, shape and orientation of the mesh elements during each time step while keeping the number of nodes and mesh topology fixed throughout the computation. As a result, the moving mesh method is also able to resolve pits which may evolve on length scales that are much smaller than the computational domain.

Continuous mesh movement approaches are often divided into two main categories: velocity-based approaches and location-based approaches. Most velocity-based or Lagrangian approaches determine the mesh movement based on the fluid or material particle flow. The Eulerian approach fixes the computational mesh and instead assumes the continuum moves with respect to the mesh nodes. In general, Eulerian meshes avoid mesh tangling and overly diffuse solutions, but the method can have difficulty resolving sharp material interfaces. The lack of advective terms in the governing equations with the Lagrangian approach make the method less diffusive than the Eulerian method. Hence, the Lagrangian approach is able to resolve sharp material interfaces [21]. A combination of Lagrangian and Eulerian approaches are used in the ALE methods [13, 14, 30, 37, 38, 55] and the ALE approach provides the basis of mesh adaptivity in COMSOL.

Location-based mesh movement directly controls the location of mesh points in particular regions of the computational domain. An example is the variational approach, which relocates the mesh points by movements derived by minimizing a functional which measures the difficulty or the error in the numerical solution [21]. Another location-based algorithm is based on an elliptic PDE descriptions. These methods can, for example, generate boundary-fitted meshes [50, 56]. This idea can be generalized using a functional [4], which provides meshes satisfying a combination of adaptivity, smoothness, and orthogonality.

Mesh adaptation functionals have been considered by many authors including the mechanical models in [22, 23, 24], vector fields in [28], a weighted Jacobian matrix approach in [29, 30], a matrix-valued diffusion coefficient in [6, 19], and the equidistribution and alignment conditions presented in [15]. The moving mesh PDE (MMPDE) method used here is based on the work in [5, 18, 18, 19, 20, 40]. Therein, a gradient flow equation determines the mesh movement, and as in the approaches above, a functional plays a vital role.

An adaptive moving mesh method, our method of choice, was proposed for relatively simple pitting corrosion problems in [41]. Our computational framework is based on the package MMPDElab by Huang [17]. MMPDElab is a general MATLAB-based adaptive moving mesh solver for time dependent PDEs where the PDEs are discretized in space using the FEM. An alternating mesh and physical solution approach is implemented to automatically obtain sufficient mesh elements in and around the evolving pit. An appropriate mesh density function, which implicitly determines an appropriate (time-dependent) distribution of elements, is constructed to track the pit growth and its interaction with any material heterogeneities.

Our test material will be stainless steel. Stainless steel is a heterogeneous, polycrystalline material where grains of differing orientations are randomly distributed through the metal [39]. In addition, further heterogeneities are invariably introduced into the structure of the steel during the steelmaking process and include defects such as voids, and metallic and non-metallic inclusions. Non-metallic inclusions such as sulphides and oxides can have widely ranging dissolution rates compared to the surrounding steel, and it has been observed that preferential dissolution can initiate corrosion pitting.

In this paper, we provide an automatic, fully adaptive simulator for pitting corrosion for arbitrary heterogeneous materials. Simulations are presented for materials containing analomous regions with varying crystallography, corrosion-resistant materials, and materials with voids.

The remainer of the paper is organized as follows. We provide a brief description of the pitting corrosion mechanism, the crystal orientation, and the associated PDE model(s) in Section 4.2. Section 4.3 provides an overview of the moving mesh methodology used in our simulations, including the specification of the mesh density function and initial mesh, as well as the FEM discretization used in MMPDElab for the physical PDE, and the overall alternating solution approach. Section 4.4 is devoted to our numerical results. The focus here is the simulation of pitting corrosion in heterogeneous materials that contain one or more regions of varying crystallography, corrosion-resistant material, or voids.

4.2 A heterogeneous PDE model for pitting corrosion

In this section, we detail our prototype model problem and adaptive solution strategy, including the description of the domain, model PDEs, boundary conditions, and necessary crystallography.

Using the conservation of mass and assuming constant diffusion, a well-mixed electrolyte and a zero net production of reactants, we arrive at the well-known Laplace's equation for the potential on the electrolyte domain, Ω, shown in Figure 4.1. Specifically, the model equations are

$$\nabla^2 \varphi = 0 \text{ in } \Omega, \tag{4.1}$$

with the following boundary conditions

$$\begin{aligned} \varphi &= 0 \text{ on } \Gamma_1, \\ \nabla\varphi \cdot \mathbf{n} &= 0 \text{ on } \Gamma_2, \Gamma_3, \Gamma_4, \\ \nabla\varphi \cdot \mathbf{n} &= \frac{i_a(\varphi)}{\sigma_c} \text{ on } \Gamma_p, \end{aligned} \tag{4.2}$$

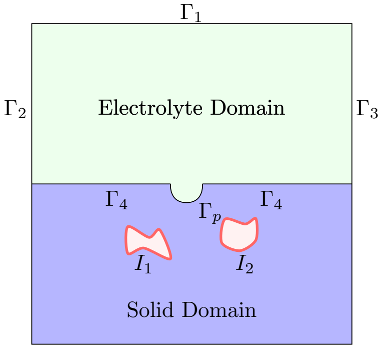


Figure 4.1: The 2D computational domain with heterogeneities.

where $\nabla\varphi \cdot \mathbf{n} = \frac{\partial\varphi}{\partial\mathbf{n}}$, $\mathbf{n}$ is the (outward) unit normal vector, $i_a(\varphi)$ is the current density, σ_c is the electrical conductivity of the electrolyte, Γ_p is the pit boundary, and Γ_1, Γ_2, Γ_3 and Γ_4 are the top, left, right and bottom of the domain, respectively. The boundary condition $\frac{\partial\varphi}{\partial\mathbf{n}} = 0$ enforced on Γ_2, Γ_3 and Γ_4 ensures there is no flow of ions across these boundaries. We denote the horizontal and vertical co-ordinates of the electrolyte region in Figure 4.1 by x and y, respectively. The shapes I_1 and I_2 represent heterogeneities in the corroding material. In practice, these could be of arbitrary number and shape.

The current density is modelled by the Butler-Volmer relation

$$i_a(\varphi) = zFA_{\text{diss}} \cdot e^{\left(\frac{zF(V_{corr}+\alpha\eta_a)}{RT}\right)}, \tag{4.3}$$

where F is Faraday's constant, z is the average charge number for the metal, α is the transfer coefficient, A_{diss} is the material dissolution affinity, T is the temperature, and R is the universal gas constant [2]. The values of these constants are recorded in Table 1. The Butler-Volmer relation is used to describe the experimental data as a

Parameter	:	Description	Value
z	:	Average charge number for the metal	2.19
F	:	Faraday's constant	96485 C/mol
R	:	Universal gas constant	8.315 J/(mol K)
T	:	Temperature	298.15 K
V_{corr}	:	Mean corrosion potential (homogeneous)	-0.24 V
V_{app}	:	Applied potential	-0.14 V
A_{diss}	:	Dissolution affinity	4 mol/cm^{2}s
C_{solid}	:	Solid concentration	143 mol/l
α	:	Transfer coefficient	0.65
Δt	:	Time step size	1

Table 4.1: List of parameters used in the corrosion model.

function of the applied over-potential

$$\eta_a = V_{app} - V_{corr} - \varphi,$$

where V_{app} and V_{corr} are the applied and the corrosion potentials, respectively [44].

As the metal corrodes, the pit becomes larger and possibly changes shape depending on the local crystallography. In our model, the new position of corrosion front, X_{new}, is computed from the old position, X_{old}, by a simple time stepping procedure

$$X_{\text{new}} = X_{\text{old}} + \Delta t V_n \mathbf{n},$$

where V_n is the magnitude of normal velocity. The magnitude of normal velocity, V_n, at the corrosion interface (or the movement of the corrosion front) is described using Faraday's law

$$V_n = \frac{i(\varphi)}{zFc_{solid}}, \tag{4.4}$$

where c_{solid} is the atomic mass concentration of the metal and z is the average charge number for the metal.

The corrosion potential as a function of crystal direction for 316 stainless steel will follow the form given by DeGiorgi et al. [32]

$$V_{corr} = k - s\left[1 - \left(\langle 001\rangle \cdot \mathbf{n}_{CD}\right)_{\max}\right], \tag{4.5}$$

where $k = -0.2297$ and $s = 0.054$. This gives a 10% difference between the maximum and minimum V_{corr} values, and further details on the application of this equation are given in [41]. A continuous spectrum of hypothetical etch rates for other materials can be obtained by varying k and s values. In addition to the simulation of the etching of stainless steel using equation (4.5), we will investigate 3 limiting cases. For simulations of a (1) homogeneous material we choose $k = -0.24$ and $s = 0$; (2) for non-etching material we set $k = -999999999$ and $0 \leq s \leq 1$; and (3) for a void $k = s = 0$.

4.3 The numerical approach

4.3.1 The adaptive moving mesh strategy

A moving mesh method automatically and continuously redistributes a fixed number of nodes where additional accuracy is required. The mesh evolves simultaneously with the solution or underlying domain by solving a MMPDE which depends on a chosen mesh density function and the (changing) boundary of the domain. The mesh density function is chosen to be large where a large mesh density is needed. This is accomplished by choosing a mesh density function which is often correlated with variations or errors in the solution of the physical PDE. If a well-resolved feature of the solution is sought at a particular location in space (here the moving pit boundary) then the mesh density function can be chosen by geometrical considerations.

Such a strategy in one spatial dimension, based on the equidistribution principle, was reviewed in paper [41]. Essentially, a mesh which equally distributes the mesh density function (in an integral sense) is found by solving a two point boundary value problem. In practice, the mesh density function depends on (features of) the solution of the physical PDE and hence the physical PDE and mesh PDE form a coupled system of equations.

In higher dimensions, we consider a mesh $\mathcal{T}_h$ of N triangular elements with N_v vertices in the physical domain $\Omega \in \mathbf{R}^d$ ($d \geq 1$). An invertible affine mapping $F_K : \hat{K} \to K$ and its Jacobian matrix, $F_K^{'}$, maps $\hat{K}$, the reference or master element, to a physical element K in $\mathcal{T}_h$. Suppose we prescribe a metric mesh tensor $\mathbb{M} = \mathbb{M}(x)$ on Ω which determines the shape, size and orientation of mesh elements of the domain Ω. A mesh is considered $\mathbb{M}$–*uniform* if all of its elements have the same size and is similar to $\hat{K}$. The main idea of the MMPDE method then is to view any adaptive mesh $\mathcal{T}_h$ as a uniform mesh in the metric $\mathbb{M}$. A standard choice of the metric tensor $\mathbb{M}$ is based on the approximate Hessian of the numerical solution. This choice of $\mathbb{M}$ is optimal in the L_2 norm of the linear interpolation error in the numerical solution of the physical PDE [16]. As discussed later, we are focused on resolving the pit geometry and hence choose $\mathbb{M}$ through these geometrical considerations.

In higher dimensions it is possible to choose $\mathbb{M}$ to control both the size and alignment of the mesh. Such a mesh is then found by minimizing a discrete functional [15] given by

$$I[\mathcal{T}_h] = \sum_{K \in \mathcal{T}_h} |K| \sqrt{\det(\mathbb{M}_K)} \left[\theta \left(\operatorname{tr}(\mathbb{J}\mathbb{M}_K^{-1}\mathbb{J}^T) \right)^{\frac{d\gamma}{2}} + (1-2\theta) d^{\frac{d\gamma}{2}} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}_K)}} \right)^{\gamma} \right], \quad (4.6)$$

where $\mathbb{J} = (F_K^{'})^{-1}$. The parameters $\theta = \frac{1}{3}$, and $\gamma = \frac{3}{2}$ are used for our numerical experiments.

This minimizing mesh can then be found by solving the gradient flow equation

$$\frac{d\mathbf{x}_i}{dt} = -\frac{P_i}{\tau} \frac{\partial I[\mathcal{T}_h]}{\partial \mathbf{x}_i}, \quad i = 1, 2, \ldots, N_v, \quad t \in (t_n, t_{n+1}], \quad (4.7)$$

where $P_i = \det(\mathbb{M}_i)^{\frac{1}{d+2}}$ is a scalar function that can be used to ensure the mesh equation has invariance properties and τ is a positive parameter used to adjust the response time of mesh movement to the change in $\mathbb{M}$.

4.3.2 Discretization and MMPDElab overview

A customized version of MMPDELab [17] is used to solve the heterogeneous corrosion problem on a grid which automatically adapts to the evolving pit geometry. A weak form of the physical PDE given by (4.1) and (4.2), is required by MMPDElab. Let V be the trial space

$$V = \{v \in H^1(\Omega(t)) : v = 0 \text{ on } \Gamma_1\} \subset H^1(\Omega(t)),$$

where $H^1(\Omega(t))$ is, roughly speaking, the function space whose members, and their first derivatives, are square integrable (see, for example, [1] for details). At any time t the weak form is constructed as follows: find $\varphi \in V$ such that

$$\int_{\Omega(t)} \nabla\varphi \cdot \nabla v d\Omega = \int_{\Gamma_p(t)} v\frac{i(\varphi)}{\sigma_c}ds, \quad \forall v \in V, \tag{4.8}$$

where $\Gamma_p(t)$ is the boundary of the pit at time t. At time t, suppose V_h is a finite dimensional subspace of V spanned by a collection basis functions associated with the mesh at that instant of time. We discretize the weak form (4.8) to find a solution in the discrete trial space. The FEM solution φ_h in the discrete trial space $V_h \subset V$ is then found by solving

$$\int_{\Omega(t)} \nabla\varphi_h \cdot \nabla v_h d\Omega = \int_{\Gamma_p(t)} \frac{i(\varphi_h)}{\sigma_c}v_h ds, \quad \forall v_h \in V. \tag{4.9}$$

We can solve the discrete variational problem (4.9) in the following way. Let $\varphi_h \in V_h$ be a linear combination of the basis functions for V_h, $\phi_j, j = 1, 2, \ldots, N$, with coefficients $\tilde{\varphi}_j$, given by

$$\varphi_h = \sum_{j=1}^{N} \tilde{\varphi}_j \phi_j. \tag{4.10}$$

Considering $v = \phi_k$, for $k = 1, 2, \ldots, N$, and using relation (4.10) gives

$$\sum_{j=1}^{N} \tilde{\varphi}_j \int_{\Omega} \nabla\phi_j.\nabla\phi_k d\Omega - \frac{1}{\sigma_c}\int_{\Gamma_p} i\Big(\sum_{j=1}^{N}\tilde{\varphi}_j\phi_j\Big)\phi_k ds = 0, \quad k = 1, 2, \ldots, N.$$

For each time, t, this system of non-linear equations is solved using Newton's method.

MMPDELab requires the user to prescribe an appropiate mesh tensor, $\mathbb{M}_K$ (see (4.6)), to precisely control how the mesh automatically adapts to the changing solution and domain features. Here, however, we wish to ensure sufficient resolution of the pit geometry using a modified distance-based monitor function given by

$$\mathbb{M}_K(x,y) = \left(1 + \frac{\mu_1}{\sqrt{\mu_2^2 d^2 + 1}}\right) I, \tag{4.11}$$

where

$$d(x,y) = \min_p |(x,y) - (x_p, y_p)|,$$

and (x_p, y_p) denotes any point on the boundary of the pit, Γ_{pit}, cf. [3]. At any point $(x,y) \in \Omega$, the value of the monitor function depends on the minimum distance, d, from (x,y) to any point on the pit boundary. $\mathbb{M}_K$ will be largest in (x,y) regions where the distance to the pit boundary is the smallest, and hence the mesh spacing will be automatically smaller in these regions. The parameters μ_1 and μ_2 control the minimum mesh spacing and the rate at which mesh clustering occurs [36]. The effect of these parameters has been evaluated in [41].

To ensure additional resolution near material heterogeneities or inclusions we will also use

$$\mathbb{M}_K(x,y) = \begin{cases} 1, & \text{if} \quad r \leq \sqrt{(x-h)^2 + (y-k)^2)}, \\ \frac{200}{e^{|r - \sqrt{(x-h)^2+(y-k)^2}|}}, & \text{if} \quad r > \sqrt{(x-h)^2 + (y-k)^2)}, \end{cases} \tag{4.12}$$

where the quantities (h,k) and r are chosen so that a circle with radius r centered at (h,k) encloses the inclusion of interest, cf. [10]. This monitor function is able to focus mesh points in a target region and a combination of (4.11) and (4.12) maybe used to resolve the required features.

The numerical simulations in this paper require the initial pit geometry and the corresponding initial spatial grid to be specified. As discussed in [41], the initial mesh on the domain may be obtained with using a tool like **initmesh** in Matlab or **mesh node** in COMSOL, or indeed using the MMPDE above. The routine **initmesh** is used in this paper, and with this initial mesh, MMPDELab uses an alternating strategy to solve the (often highly nonlinear) coupled physical PDEs and mesh equation forward in time.

The direction of pit growth is normal to the pit boundary and the magnitude of the normal velocity of each vertex on the boundary of the pit is given by (4.4). The positions of any corner nodes are moved as described in [41].

4.4 Numerical results

Throughout this section a constant value of $\tau = 10^{-5}$ is used in the MMPDE, and the constants $\mu_1 = 100$ and $\mu_2 = 1$ are used as default values for the mesh density function (4.11). The initial number of mesh points on the boundary of the pit is set to 61.

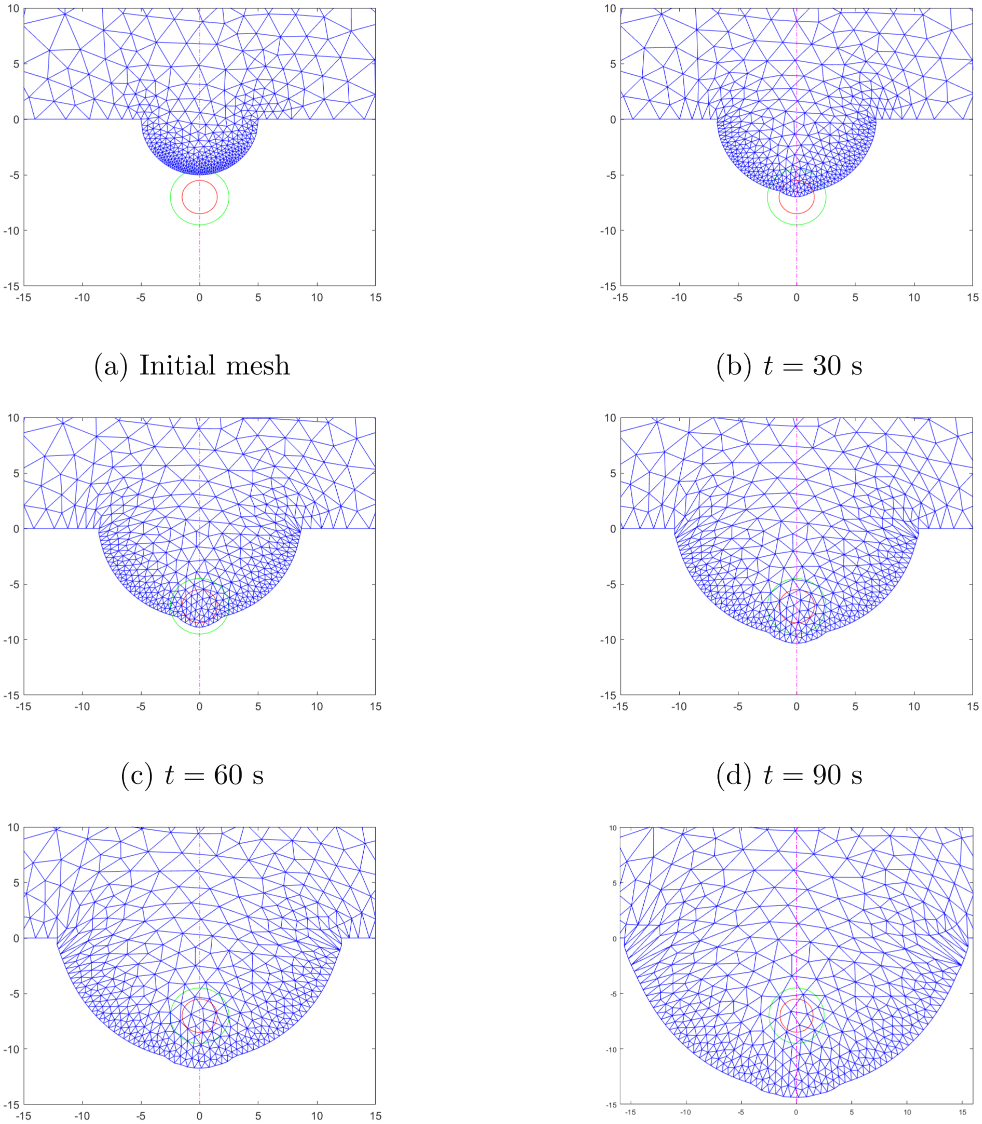
4.4.1 Case I: Inclusion(s) with different crystallography

We begin by considering a single circular crystalline inclusion centered at $(0, -7)$ with radius 1.5 and crystal direction [001] as defined by the zone axis of the inclusion. The location and size of the inclusion are highlighted by the red circles in Figure 4.2. For these simulations, we use a combination of two monitor functions. First, the distance based monitor function (4.11) is applied to track the pit boundary. Once the pit boundary reaches the green circle around the inclusion (with radius 2.5) a mesh solve using the distance-based monitor function is followed by a mesh solve using the exponential-based monitor function (4.12) in order to get increased resolution of the mesh near (and around) the inclusion. The sequential use of the two monitor functions continues as long as the pit boundary lies within the green circle. Presented in Figure 4.2 are meshes at 6 time steps ranging from 0 s to 180 s. It is observed that the circular inclusion etches faster than the surrounding homogeneous material as highlighted by the protrusion of the pit boundary at $x = 0$, and this effect can be attributed to the difference in V_{corr} values between the two materials. The V_{corr} value for the inclusion is -0.2294 V, as calculated from equation (9), and for the surrounding homogeneous material V_{corr} is -0.24 V. Hence, the inclusion will have a greater normal velocity of pit boundary movement as indicated by equations (4.3) to (4.5). Once the lower boundary of the inclusion has been reached, pit boundary movement returns to the behaviour of the homogeneous material over the entire boundary. In addition, it is observed that the pit boundary near the protrusion smooths out with time due to averaging of the V_{corr} values at the boundary nodes.

Some stretching of the elements is noticeable at the corners of the pit by $t = 90$ s. This effect can be reduced by using a monitor function which will be larger near the pit corners. Alternatively, a periodic h-refinement could be used to increase the total number of mesh points which will be the focus of future study, cf. [26].

We now consider a pit encountering two circular inclusions located at (-3, -6.5) and (3, 6.5) as shown in Figure 4.3. Unlike the geometry for a single inclusion shown in Figure 2, where the bottom of the pit boundary contacted the top of the inclusion with a crystal direction of [010], the sides of the pit boundary contact close to the side of the inclusion orientated along [$\bar{1}$10]. The [$\bar{1}$10] direction of the inclusion has a V_{corr} value of -0.2455 V, a magnitude about 2.3% larger than the surrounding homogeneous material, and we would expect the inclusion to etch a little slower than the adjacent homogenous material. As the pit boundary outside of the inclusion travels faster than inside the inclusion, it will contact the inclusion and then slow down. In fact, it is expected that the rate of boundary movement within the inclusion will equilibrate along the crystallographic direction with the lowest rate, which in this case is [$\bar{1}$10], thereby producing a crystallographic facet on the pit boundary along the ($\bar{1}$10) plane.

The remaining features to be explained for the two-inclusion case are the two protrusions located at the inner edges of the inclusion closest to the pit boundary at $x = 0$, as evident in Figure 4.3 for times 60 s and greater. The pit boundary movement velocity varies with direction within the crystal, and plotted in Figure 4.4(a) are the V_{corr} values as a function of crystallographic direction, where ‘Faster’ and ‘Slower’ denote pit boundary velocities within the inclusion that are faster and slower than the surrounding homogeneous material, respectively. It is observed that only for a narrow range of directions about [$\bar{1}$10] is the boundary movement within the inclusion slower than the homogeneous material. Outside of this region, the velocity



(a) Initial mesh

(b) $t = 30$ s

(c) $t = 60$ s

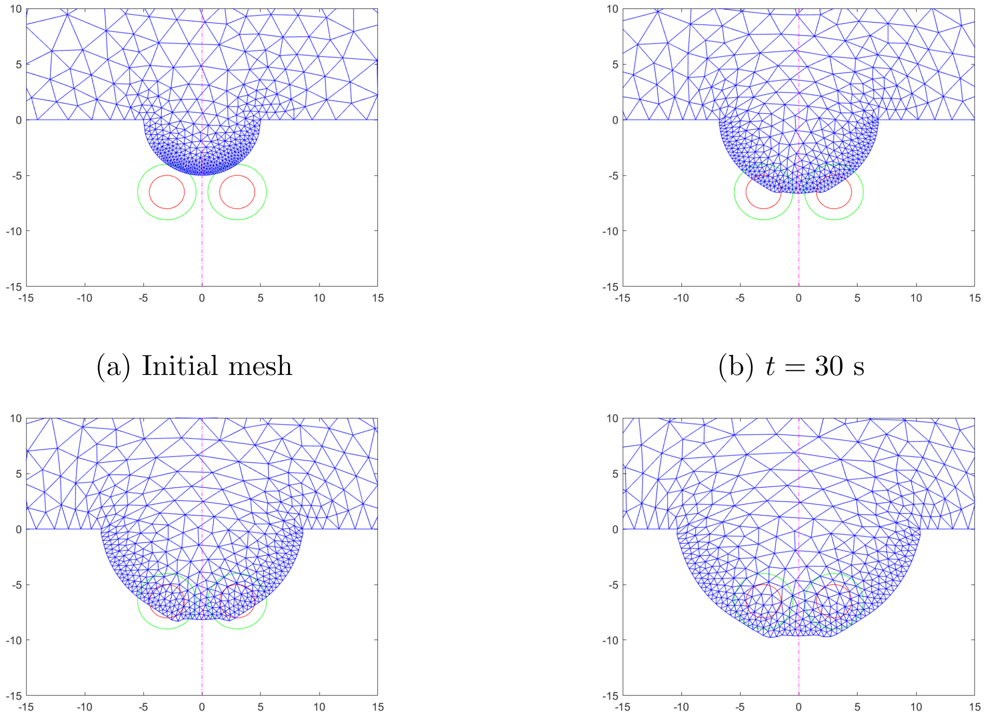
(d) $t = 90$ s

(e) $t = 120$ s

(f) $t = 180$ s

Figure 4.2: Meshes for a corrosion pit at various times as the pit encounters a single crystalline inclusion (outlined by the red circles) with crystal direction [001]. The green circles outline the locations where the monitor function changes to achieve greater node resolution at the surface of the inclusion.

of the pit boundary movement is greater. Now, consider Figure 4.4(b) which displays the crystallographic directions of the inclusion with respect to the pit boundary. The



(a) Initial mesh

(b) $t = 30$ s

(c) $t = 60$ s

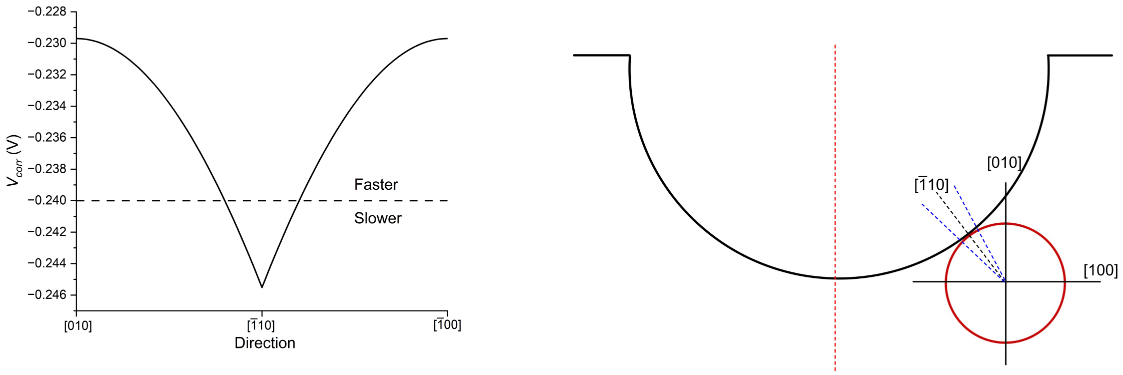
(d) $t = 90$ s

Figure 4.3: Meshes at various times for a pit encountering two circular inclusions with crystal direction [001].

region within the cone centered about $[\bar{1}10]$ will have velocities less than the homogeneous material. As the pit boundary proceeds through the inclusion, the boundary to the left of the cone will move faster than the surrounding homogenous material for a period of time before slowing as it approaches the $(\bar{1}10)$ plane causing the protrusion.

4.4.2 Case II: Corrosion-resistant inclusions

Corrosion resistance is achieved in the model by setting the k value in equation (4.5) to a very large negative number so that the current density within the inclusion is effectively zero, leading to a normal velocity of the pit boundary of also zero. As was the case for the crystalline inclusions, single and double corrosion-resistant inclusions

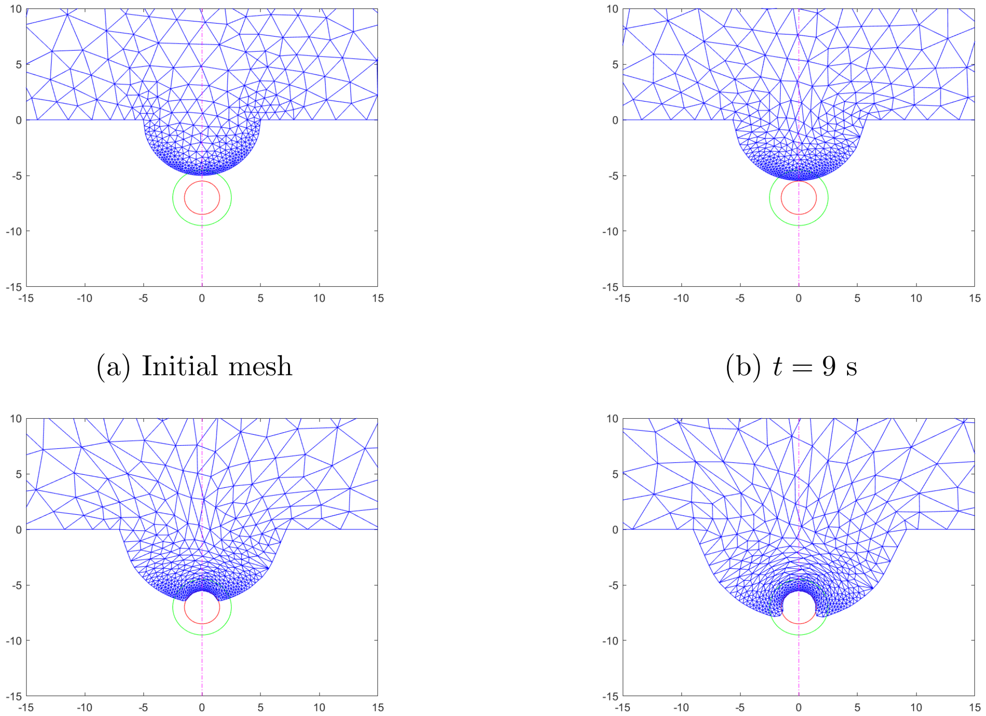


(a)

(b)

Figure 4.4: (a) Plot of V_{corr} as a function of direction within the crystalline inclusion. More negative V_{corr} values lead to slower movement of the pit boundary. (b) A diagram displaying the relationship of the crystallographic directions of the inclusion with respect to the pit boundary.

with a radius of 1.5 were studied. The single inclusion was located at the bottom of the pit at coordinates (0, -7), and the two-inclusion simulations had inclusions located at (-3, -6.5) and (3, 6.5). The same meshing strategy as for the crystalline inclusions was followed, and mesh plots at various times for the single and double inclusion cases are shown in Figures 4.5 and 4.6, respectively. In both cases, this meshing procedure performs well up to t = 30 s, but by t = 60 s the mesh quality degrades, particularly for the double inclusion case where the mesh appears inside one of the inclusions. By definition, the material within a corrosion-resistant inclusion should not be removed during the corrosion process, and this is a serious artefact. The reason for this behaviour is low mesh density right at the leading edge of the boundary adjacent to the inclusion. A meshing strategy involving h-refinement in addition to the r-refinement presently being used is expected to improve the mesh quality around the pit and is presently under investigation.



(a) Initial mesh

(b) $t = 9$ s

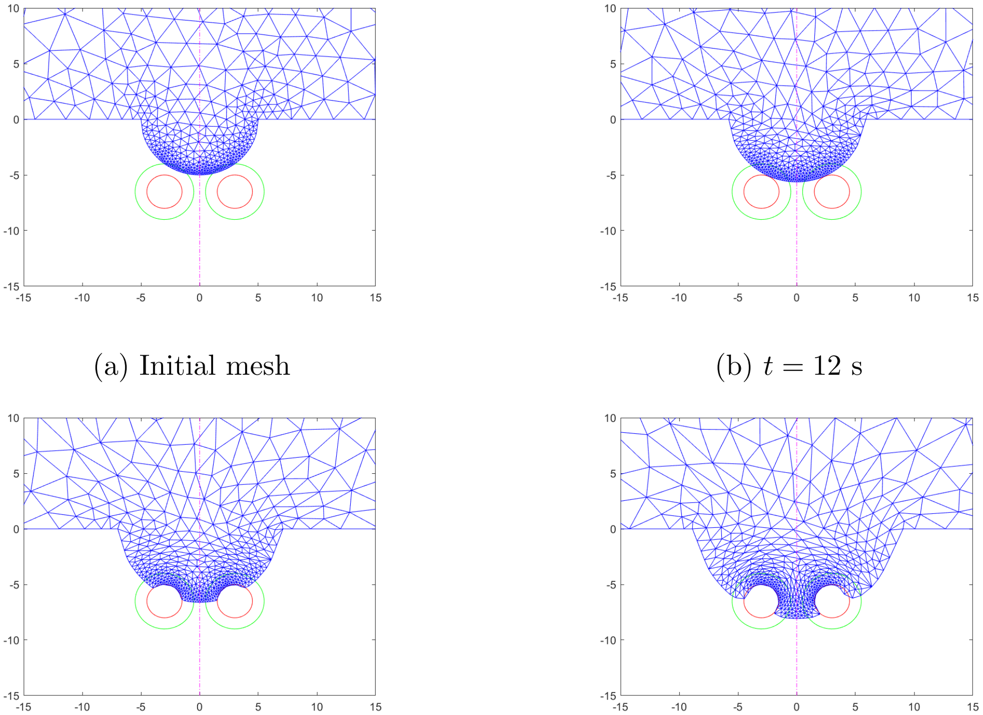
(c) $t = 30$ s

(d) $t = 60$ s

Figure 4.5: Meshes at various times as a pit encounters a single circular corrosion-resistant inclusion outlined in red. The surrounding green circle indicates the region where the monitor function changes from purely distance-based (4.11) to a sequence of distance- and exponential-based (4.12) monitor functions to increase mesh density at the inclusion.

4.4.3 Case III: A pit encountering material voids

At the other extreme from a corrosion-resistant inclusion is a void in the material, which can be common in stainless steels. Voids present a unique challenge since there is a topography change when the pit boundary encounters the void. Moving mesh methods are not able to continuously handle the abrupt topography change, instead the electrolyte domain must be remeshed prior to continuation of the adaptive moving mesh procedure. The remeshing is triggered when any pit boundary edge encounters the void.



(a) Initial mesh

(b) $t = 12$ s

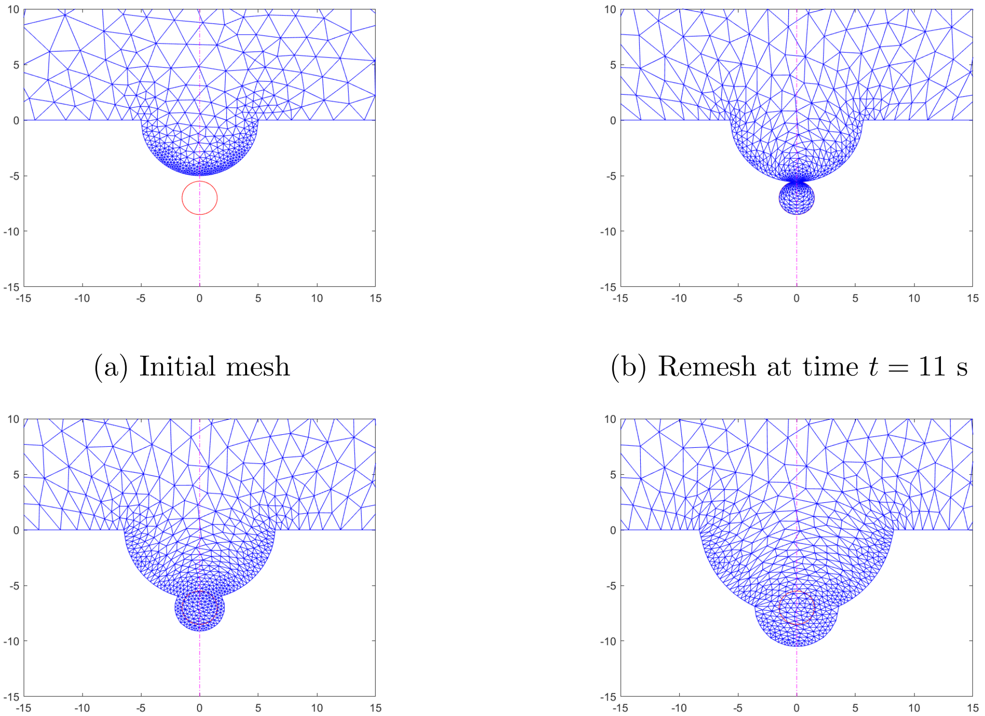
(c) $t = 30$ s

(d) $t = 60$ s

Figure 4.6: Meshes at various times as a pit encounters multiple corrosion resistant inclusions.

The single void was located at (0, -7), and in the two void case the voids were located at (0, -7) and (6, -7). The two voids were not symmetrically located about the pit in order to trigger two topology changes requiring remeshing as a test of the robustness of the technique. Only the distance-based monitor function was used to position the nodes on the pit boundary.

Figures 4.7 and 4.8 display the meshes for single and double voids, respectively. In both cases, the first void was detected at $t = 11$ s, triggering a remeshing operation using **initmesh**. A high density of nodes is placed within the void and at the point where the pit boundary and node touched. Time stepping proceeded as per normal, with the nodes redistributing as required by the distance-based monitor function. In the case of two voids, the second void intersected the pit boundary at $t = 60$, triggering another remeshing, and this technique can be extended to an arbitrary number of voids with varying shapes.



(a) Initial mesh

(b) Remesh at time $t = 11$ s

(c) $t = 30$ s

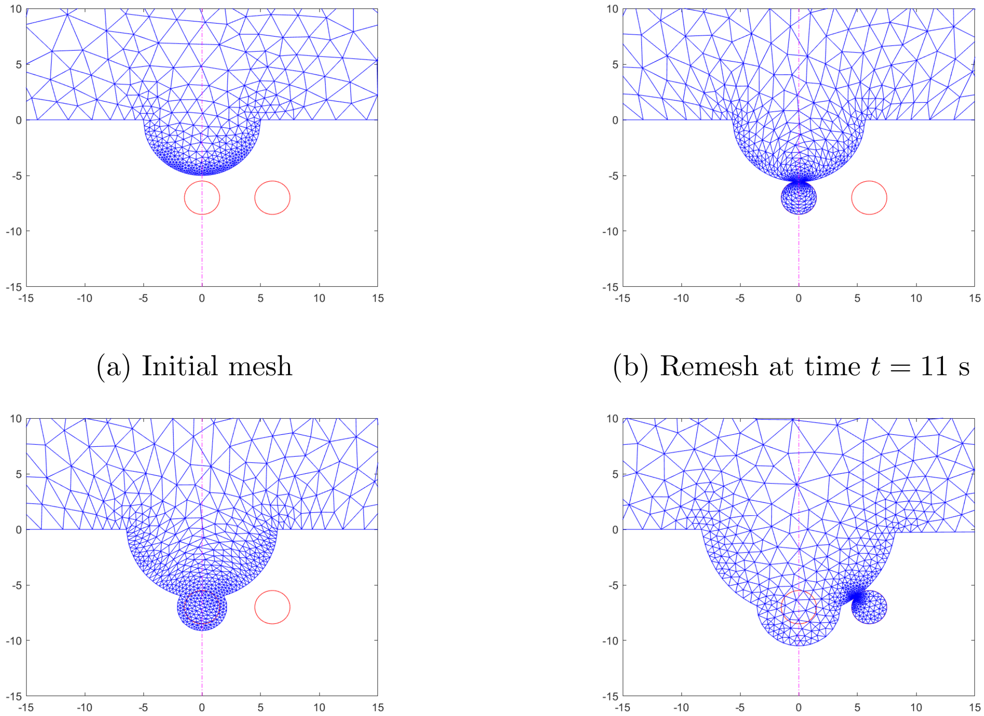
(d) $t = 60$ s

Figure 4.7: Meshes at various times as a pit encounters a single material void.

4.5 Conclusion

This paper presents a robust, fully automatic, moving mesh solution framework for pitting corrosion in heterogeneous materials or situations where material voids are present. The moving mesh approach is able to continuously and smoothly evolve a fixed mesh topology according to the changing pit geometry as the pits encounter inclusions with varying crystallography or voids.

Designing mesh density functions capable of resolving corrosion dynamics in situations with several competing regions of interest is difficult and would certainly benefit



(a) Initial mesh

(b) Remesh at time $t = 11$ s

(c) $t = 30$ s

(d) Second remesh at time $t = 60$ s

Figure 4.8: Meshes at various times as a pit encounters multiple circular material voids.

from an hr-refinement strategy, which both redistributes nodes as we have presented here, but also allows periodic changes to the number of mesh nodes. This work is ongoing and will appear in a subsequent paper.

4.6 Data availability

The raw or processed data required to reproduce these figures and findings are available from the authors upon request.

4.7 Acknowledgements

The work of Ronald D. Haynes and Abu N. Sarker (PhD student) has been supported by a NSERC discovery grant (Canada).

Bibliography

[1] David R Adams and Lars I Hedberg. *Function Spaces and Potential Theory*, volume 314. Springer Science & Business Media, 1999.

[2] Allen J Bard, Larry R Faulkner, and Henry S White. *Electrochemical Methods: Fundamentals and Applications*. John Wiley & Sons, 2022.

[3] G Beckett, John A Mackenzie, and ML Robertson. A moving mesh finite element method for the solution of two-dimensional Stefan problems. *Journal of Computational Physics*, 168(2):500–518, 2001.

[4] Jeremiah U Brackbill and Jeff S Saltzman. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics*, 46(3):342–368, 1982.

[5] Weiming Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM Journal on Numerical Analysis*, 43(1):19–40, 2005.

[6] Weiming Cao, Weizhang Huang, and Robert D Russell. A study of monitor functions for two-dimensional adaptive mesh generation. *SIAM Journal on Scientific Computing*, 20(6):1978–1994, 1999.

[7] BW Davis, PJ Moran, and PM Natishan. Metastable pitting behavior of aluminum single crystals. *Corrosion Science*, 42(12):2187–2192, 2000.

[8] Dennj De Meo and Erkan Oterkus. Finite element implementation of a peridynamic pitting corrosion damage model. *Ocean Engineering*, 135:76–83, 2017.

[9] Virginia G DeGiorgi, Nithyanand Kota, Alexis C Lewis, and Siddiq M Qidwai. Numerical modeling of pit growth in microstructure. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55850, pages 1–7. American Society of Mechanical Engineers, 2013.

[10] Kelsey L DiPietro, Ronald D Haynes, Weizhang Huang, Alan E Lindsay, and Yufei Yu. Moving mesh simulation of contact sets in two dimensional models of elastic–electrostatic deflection problems. *Journal of Computational Physics*, 375:763–782, 2018.

[11] Ravindra Duddu. Numerical modeling of corrosion pit propagation using the combined extended finite element and level set method. *Computational Mechanics*, 54(3):613–627, 2014.

[12] MA Hill, JF Bingert, and RS Lillard. The relationship between crystallographic orientation and the passivity and breakdown of beryllium. In *194th Meeting of the Electrochemical Society*. 1998.

[13] CW Hirt, AA Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.

[14] Cyrill W Hirt, Anthony A Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[15] Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *Journal of Computational Physics*, 174(2):903–924, 2001.

[16] Weizhang Huang. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics*, 204(2):633–665, 2005.

[17] Weizhang Huang. An introduction to MMPDElab. *ArXiv Preprint ArXiv:1904.05535*, 2019.

[18] Weizhang Huang, Yuhe Ren, and Robert D Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM Journal on Numerical Analysis*, 31(3):709–730, 1994.

[19] Weizhang Huang and Robert D Russell. A high dimensional moving mesh strategy. *Applied Numerical Mathematics*, 26(1-2):63–76, 1998.

[20] Weizhang Huang and Robert D Russell. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM Journal on Scientific Computing*, 20(3):998–1015, 1998.

[21] Weizhang Huang and Robert D Russell. *Adaptive Moving Mesh Methods*, volume 174. Springer Science & Business Media, 2010.

[22] Olivier-Pierre Jacquotte. A mechanical model for a new grid generation method in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 66(3):323–338, 1988.

[23] Olivier-Pierre Jacquotte. Generation, optimization and adaptation of multiblock grids around complex configurations in computational fluid dynamics. *International Journal for Numerical Methods in Engineering*, 34(2):443–454, 1992.

[24] Olivier-Pierre Jacquotte and Grégory Coussement. Structured mesh adaption: space accuracy and interpolation methods. *Computer Methods in Applied Mechanics and Engineering*, 101(1-3):397–432, 1992.

[25] Siavash Jafarzadeh, Ziguang Chen, and Florin Bobaru. Computational modeling of pitting corrosion. *Corrosion Reviews*, 37(5):419–439, 2019.

[26] Hormoz Jahandari, Scott MacLachlan, Ronald D. Haynes, and Niall Madden. Finite element modelling of geophysical electromagnetic data with goal-oriented *hr*-adaptivity. *Computational Geosciences*, 24(3):1257–1283, 2020.

[27] SMLC Jain, MLC Lim, JL Hudson, and JR Scully. Spreading of intergranular corrosion on the surface of sensitized Al-4.4 mg alloys: A general finding. *Corrosion Science*, 59:136–147, 2012.

[28] Patrick M Knupp. Mesh generation using vector fields. *Journal of Computational Physics*, 119(1):142–148, 1995.

[29] Patrick M Knupp. Jacobian-weighted elliptic grid generation. *SIAM Journal on Scientific Computing*, 17(6):1475–1490, 1996.

[30] Patrick M Knupp, Len G Margolin, and Mikhail Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176(1):93–128, 2002.

[31] EV Koroleva, GE Thompson, P Skeldon, and B Noble. Crystallographic dissolution of high purity aluminium. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 463, pages 1729–1748. The Royal Society, 2007.

[32] Nithyanand Kota, Siddiq M Qidwai, Alexis C Lewis, and Virginia G DeGiorgi. Microstructure-based numerical modeling of pitting corrosion in 316 stainless steel. *ECS Transactions*, 50(31):155–164, 2013.

[33] D Krouse, N Laycock, and C Padovani. Modelling pitting corrosion of stainless steel in atmospheric exposures to chloride containing environments. *Corrosion Engineering, Science and Technology*, 49(6):521–528, 2014.

[34] B Ravi Kumar, Raghuvir Singh, Bhupeshwar Mahato, PK De, NR Bandyopadhyay, and DK Bhattacharya. Effect of texture on corrosion behavior of aisi 304l stainless steel. *Materials Characterization*, 54(2):141–147, 02 2005.

[35] Nicholas J Laycock, Donal P Krouse, Shaun C Hendy, and David E Williams. Computer simulation of pitting corrosion of stainless steels. *The Electrochemical Society Interface*, 23(4):65–71, 2014.

[36] JA Mackenzie and ML Robertson. The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method. *Journal of Computational Physics*, 161(2):537–557, 2000.

[37] Len G Margolin. Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):198–202, 1997.

[38] Fabio Nobile and Luca Formaggia. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-West Journal of Numerical Mathematics*, 7:105–132, 1999.

[39] Joo Hyun Park and Youngjo Kang. Inclusions in stainless steels - A review. *Steel Res. Int.*, 88(12):1700130/1–26, dec 2017.

[40] Yuhe Ren and Robert D Russell. Moving mesh techniques based upon equidistribution, and their stability. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1265–1286, 1992.

[41] Abu N. Sarker, Ronald D. Haynes, and Michael Robertson. Moving mesh simulations of pitting corrosion, 2023. Preprint, Submitted.

[42] Akinori Sato, Kenzo Kon, Shigeo Tsujikawa, and Yoshihiro Hisamatsu. Effect of crystallographic orientation on dissolution behavior of stainless steels single crystal. *Materials Transactions, JIM*, 37(4):729–732, 1996.

[43] Stefan Scheiner and Christian Hellmich. Stable pitting corrosion of stainless steel as diffusion-controlled dissolution process with a sharp moving electrode boundary. *Corrosion Science*, 49(2):319–346, 2007.

[44] Stefan Scheiner and Christian Hellmich. Finite volume model for diffusion-and activation-controlled pitting corrosion of stainless steel. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2898–2910, 2009.

[45] Jong Hyun Seo, Jong-Ho Ryu, and Dong Nyung Lee. Formation of crystallographic etch pits during AC etching of aluminum. *Journal of the Electrochemical Society*, 150(9):B433–B438, 2003.

[46] SM Sharland. A review of the theoretical modelling of crevice and pitting corrosion. *Corrosion Science*, 27(3):289–323, 1987.

[47] SM Sharland. A mathematical model of crevice and pitting corrosion—II. the mathematical solution. *Corrosion Science*, 28(6):621–630, 1988.

[48] SM Sharland and PW Tasker. A mathematical model of crevice and pitting corrosion-I. The physical model. *Corrosion Science*, 28(6):603–620, 1988.

[49] Laura B Simon, Mohammad Khobaib, Theodore E Matikas, CS Jeffcoate, and MS Donley. Influence of pitting corrosion on structural integrity of aluminum alloys. In *Nondestructive Evaluation of Aging Materials and Composites III*, volume 3585, pages 40–47. International Society for Optics and Photonics, 1999.

[50] Joe F Thompson, Frank C Thames, and C Wayne Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, 15(3):299–319, 1974.

[51] Grainne M Treacy and Carmel B Breslin. Electrochemical studies on single-crystal aluminium surfaces. *Electrochimica Acta*, 43(12):1715–1720, 1998.

[52] A Turnbull, DA Horner, and BJ Connolly. Challenges in modelling the evolution of stress corrosion cracks from pits. *Engineering Fracture Mechanics*, 76(5):633–640, 2009.

[53] John C Walton. Mathematical modeling of mass transport and chemical reaction in crevice and pitting corrosion. *Corrosion Science*, 30(8-9):915–928, 1990.

[54] Kai Wang, Chenpei Li, Yanhui Li, Jinling Lu, Yueshe Wang, and Xingqi Luo. Multi-physics coupling analysis on the time-dependent localized corrosion behavior of carbon steel in CO_2-H_2O environment. *Journal of The Electrochemical Society*, 167(1):013505–013505, 2019.

[55] BV Wells, Michael J Baines, and Paul Glaister. Generation of arbitrary Lagrangian–Eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations. *International Journal for Numerical Methods in Fluids*, 47(10-11):1375–1381, 2005.

[56] Alan M Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.

[57] M Yasuda, F Weinberg, and D Tromans. Pitting corrosion of Al and Al-Cu single crystals. *Journal of the Electrochemical Society*, 137(12):3708–3715, 1990.

[58] Xingliang Yu, Fangyi Wan, and Yingnan Guo. Micromechanics modeling of skin panel with pitting corrosion for aircraft structural health monitoring. In *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE, 2016.

Chapter 5

An adaptive hr-refinement simulation for pitting corrosion of heterogeneous materials

Abstract

[1] The design and implementation of an hr–refinement method is provided for the simulation of pitting corrosion for materials with heterogeneous inclusions. The adaptive mesh is generated automatically by alternating mesh PDE solves (which are coupled to a nonlinear potential problem) and h-refinements. h-refinement is triggered once an appropriate mesh quality measure exceeds a prescribed tolerance. The result is a new extension of MMPDElab which provides hr–refinement. In the context of pitting corrosion, the method is able to resolve pit growth in complex domains corresponding to heterogeneous materials. The software is able to smoothly tackle the changing pit geometry associated with materials with inclusions of varying crystallography and corrosion resistance.

Keywords: Pitting Corrosion, Adaptive Moving Mesh, MMPDE, FEM, h-refinement,

[1] This work is under revision as "*An adaptive hr-refinement simulation for pitting corrosion of heterogeneous materials*", by A. N. Sarker, R. D. Haynes, and M. D. Robertson, 2023, Preprint.

Crystallography, Heterogeneous Materials

5.1 Introduction

Damage due to pitting corrosion of metals cost governments and industries billions of dollars per year and can put human lives at risk. It is a particularly insidious, localized form of corrosion which generates small pits, cavities, or holes in metal. Pits may be small on the surface but larger below, making physical detection quite difficult. Hence, the structural integrity of metal may be impacted before it is observed [34, 41]. The shape and size of a pit depend on the components of the metal, the orientation of metal surface, and the local chemical environment [31].

Pitting corrosion has two main phases: an initiation phase and a growth phase. The initiation of corrosion pits on various metals has been studied including stainless steel [7, 20, 27], beryllium [7], pure aluminum and aluminum alloys [15, 17, 30, 35, 40]. Experimental studies of pit initiation and growth is often time consuming, expensive, and physically difficult in many situations. Mathematical modelling and numerical simulations can supplement the experimentation of corrosion for a wide range of materials and environmental conditions. In this study, our test material is 316 stainless steel as described by the parameters provided in [5, 18].

Recently, [4, 5, 6, 18, 19, 21, 28, 29, 29, 32, 33, 36, 37, 38] have developed finite element (FEM) or finite volume approaches for partial differential equation (PDE) based models of pitting corrosion. An extensive overview discusses of models of the anodic reaction at the corrosion front and the transportation of ions in the electrolyte [13]. In many of these papers, the commercial software COMSOL® has been used to solve the PDE model in the electrolyte domain.

As we will see in Section 5.2, the model of interest is a moving boundary problem due to the growth of the electrolyte domain as the corrosion proceeds. Numerical approximations of the solution of the moving boundary problem necessitates a remeshing of the time dependent computational domain at each time step. The COMSOL® implementations mentioned above utilizes an adaptive meshing routine based on an arbitrary Lagrangian-Eulerian (ALE) approach [8, 9, 16, 23, 24, 39]. An alternative approach is considered here. An adaptive moving mesh method which allows us to automatically and continuously vary the size, shape and orientation of the mesh elements, while keeping the number of nodes and mesh topology fixed throughout the computation was used in [25, 26]. By specifying an appropriate mesh density function, this approach was shown to give fine (automatic) control over the quality of the underlying mesh throughout the computation for many test situations. For heterogeneous materials, however, there was some difficulty in prescribing a mesh density function which was capable of balancing the competing features of the solution. Moreover, it is clear that as the pit continues to grow, the fixed number of nodes in the moving mesh approach will not be sufficient to resolve the quantities of interest to the required accuracy.

The moving mesh approach used in this paper is a location-based method which relocates (the r in r-refinement) the mesh points in particular regions of the computational domain by minimizing a functional which measures the difficulty or the error in the numerical solution [12]. Our solver builds upon the general moving mesh PDE (MMPDE) approach provided by the software MMPDElab [11]. Therein, a gradient flow equation is used to find the minimizer of the prescribed functional.

To deal with the stress that is applied to a r-refinement approach by a growing corrosion pit in a heterogeneous material, here we couple the r-refinement strategy with h-refinement that adds mesh elements automatically in regions with poor grid quality. h-refinement is a standard approach in FEM simulation, usually coupled in a hp-refinement strategy which automatically controls in the number of elements and the order of the FEM approximation as the simulation progresses. Here we couple h-refinement with r-refinement to provide an hr-refinement solver (cf. [14]) for pitting corrosion.

In this paper, we provide an automatic, fully adaptive simulator for pitting corrosion generalizable to arbitrary heterogeneous materials. By providing appropriate mesh density functions, simulations of pit growth are presented for materials containing anomalous regions with varying crystallography, corrosion resistant materials, and materials with voids.

The remainder of the paper is organized as follows. We provide a brief description of the pitting corrosion mechanism, the crystal orientation, and the associated PDE-model(s) in Section 5.2. Section 5.3 provides an overview of the moving mesh methodology used in our simulations including the specification of the mesh density functions, the FEM method used in MMPDElab for the physical PDE, the boundary movement strategy, and the overall alternating solution approach. Section 5.4 is devoted to our numerical results.

5.2 A heterogeneous PDE model for pitting corrosion

This section provides our prototype model problem including a description of the domain, the model PDEs and boundary conditions, and the necessary crystallography. Using the conservation of mass and assuming a constant diffusion, a well-mixed electrolyte and a zero net production of reactants, we arrive at the well-known Laplace's equation for the electrical potential on the electrolyte domain, Ω, as shown in Figure 5.1. Specifically, the model equations are

$$\nabla^2\varphi = 0 \text{ in } \Omega, \tag{5.1}$$

with the following boundary conditions

$$\begin{aligned}\varphi &= 0 \text{ on } \Gamma_1,\\ \nabla\varphi\cdot\mathbf{n} &= 0 \text{ on } \Gamma_2,\Gamma_3,\Gamma_4,\\ \nabla\varphi\cdot\mathbf{n} &= \frac{i_a(\varphi)}{\sigma_c} \text{ on } \Gamma_p,\end{aligned} \tag{5.2}$$

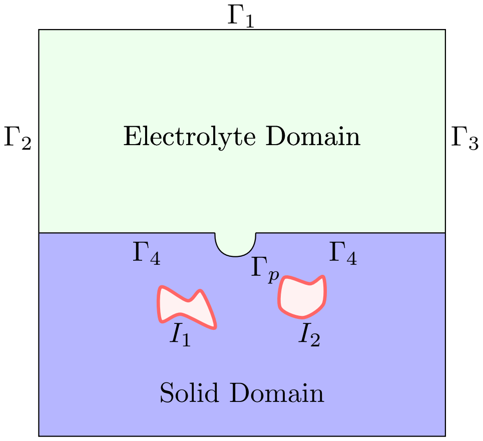


Figure 5.1: The 2D computational domain with heterogeneities.

where $\nabla\varphi\cdot\mathbf{n} = \frac{\partial\varphi}{\partial\mathbf{n}}$, $\mathbf{n}$ is the (outward) unit normal vector, $i_a(\varphi)$ is the current density,

σ_c is the electrical conductivity of the electrolyte, Γ_p is the pit boundary, and Γ_1, Γ_2, Γ_3 and Γ_4 are the top, left, right and bottom of the domain, respectively. No flow of ions is permitted across Γ_2, Γ_3 and Γ_4. This is enforced by the boundary condition $\frac{\partial \varphi}{\partial \mathbf{n}} = 0$. The horizontal and vertical co-ordinates of the electrolyte region in Figure 5.1 are labelled x and y, respectively. The shapes I_1 and I_2 represent heterogeneities in the corroding material.

The Butler-Volmer relation models the current density as

$$i_a(\varphi) = zFA_s \cdot e^{\left(\frac{zF(V_{corr}+\alpha\eta_a)}{RT}\right)}, \tag{5.3}$$

where F is Faraday's constant, z is the average charge number for the metal, α is the transfer coefficient, A_{diss} is the material dissolution affinity, T is the temperature and R is the universal gas constant [2]. The values of these constants are recorded in Table 1. The applied over-potential is given as

$$\eta_a = V_{app} - V_{corr} - \varphi,$$

where V_{app} and V_{corr} are the applied and the corrosion potentials, respectively [29].

As the metal corrodes, the pit becomes larger and possibly changes shape as the metal corrodes. In our model the new position of the corrosion front, X_{new}, is computed from the old position, X_{old}, by a simple time stepping procedure

$$X_{\text{new}} = X_{\text{old}} + \Delta t V_n \mathbf{n},$$

where V_n is the magnitude of normal velocity at the corrosion interface given by Faraday's law

$$V_n = \frac{i(\varphi)}{zFc_{solid}}. \tag{5.4}$$

Here c_{solid} is the atomic mass concentration and z is the average charge number of the metal. The values of these parameters are summarized in Table 5.1.

DeGiorgi et al. [18] gives the corrosion potential as a function of crystal direction for 316 stainless steel as

$$V_{corr} = k - s\left[1 - \left(\langle 001\rangle \cdot \mathbf{n}_{CD}\right)_{\max}\right], \tag{5.5}$$

Parameter	:	Description	Value
z	:	Average charge number for the metal	2.19
F	:	Faraday's constant	96485 C/mol
R	:	Universal gas constant	8.315 J/(mol K)
T	:	Temperature	298.15 K
V_{corr}	:	Mean corrosion potential (homogeneous)	-0.24 V
V_{app}	:	Applied potential	-0.14 V
α	:	Transfer coefficient	0.65
A_{diss}	:	Dissolution affinity	4 mol/cm^{2}s
C_{solid}	:	Solid concentration	143 mol/l
Δt	:	Time step size	1

Table 5.1: List of parameters used in the corrosion model.

where $k = -0.2297$ and $s = 0.054$. This gives a 10% difference between the maximum and minimum V_{corr} values. See [26] for further details on the application of this equation. A continuous spectrum of hypothetical etch rates for other materials can be obtained by varying the k and s values. We will illustrate our hr-refinement approach using various limiting cases: (1) a homogeneous material with $k = -0.24$ and $s = 0$; (2) a non-etching material with $k = -999999999$; and (3) a void or hole with $k = s = 0$.

5.3 The numerical approach

5.3.1 The adaptive moving mesh strategy

Moving mesh methods automatically and continuously redistribute a fixed number of nodes where additional accuracy is required. The mesh is obtained by solving a MMPDE which depends on a chosen mesh density function and the possibly time dependent computational domain. The mesh density function is often correlated with estimated errors in the approximate solution of the physical PDE or by geometrical considerations.

In two dimensions, we consider a mesh $\mathcal{T}_h$ of N triangular elements with N_v vertices in the physical domain $\Omega \in \mathbf{R}^2$. An invertible affine mapping $F_K : \hat{K} \to K$ and its Jacobian matrix, $F_K^{'}$, maps $\hat{K}$, the reference or master element, to a physical element K in $\mathcal{T}_h$. A metric mesh tensor $\mathbb{M} = \mathbb{M}(x)$ is prescribed on Ω which determines the shape, size and orientation of mesh elements of the domain Ω. A mesh is considered $\mathbb{M}$–uniform if all of its elements have the same size and is similar to $\hat{K}$. The adaptive mesh $\mathcal{T}_h$ obtained by the MMPDE approach is a uniform mesh in the metric $\mathbb{M}$. As our goal is automatic resolution of pit geometry, we choose $\mathbb{M}$ based on geometrical considerations.

It is possible [10] to construct $\mathbb{M}$ to control both the size and alignment of the mesh, and then minimize,

$$I[\mathcal{T}_h] = \sum_{K \in \mathcal{T}_h} |K| \sqrt{\det(\mathbb{M}_K)} \left[\theta \left(\operatorname{tr}(\mathbb{J}\mathbb{M}_K^{-1}\mathbb{J}^T) \right)^{\frac{d\gamma}{2}} + (1-2\theta) d^{\frac{d\gamma}{2}} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}_K)}} \right)^{\gamma} \right], \quad (5.6)$$

where $\mathbb{J} = (F_K^{'})^{-1}$. The parameters $\theta = \frac{1}{3}$, and $\gamma = \frac{3}{2}$ are used for our numerical experiments.

This functional is minimized by integrating the gradient flow equation

$$\frac{d\mathbf{x}_i}{dt} = -\frac{P_i}{\tau} \frac{\partial I[\mathcal{T}_h]}{\partial \mathbf{x}_i}, \quad i = 1, 2, \ldots, N_v, \quad t \in (t_n, t_{n+1}], \quad (5.7)$$

where $P_i = \det(\mathbb{M}_i)^{\frac{1}{d+2}}$ ensures that the mesh equation has invariance properties and τ adjusts the response time of mesh movement to the change in $\mathbb{M}$.

5.3.2 Discretization and MMPDElab overview

The moving boundary problem associated with pitting corrosion is solved using a customized version of MMPDELab [11] on a grid which automatically adapts to the evolving pit geometry and heterogeneities. As is typical for FEM simulations, MMPDElab requires a weak form of the physical PDE (5.1) and (5.2).

At any time t the weak form is constructed as follows: find $\varphi \in V$ such that

$$\int_{\Omega(t)} \nabla\varphi \cdot \nabla v d\Omega = \int_{\Gamma_p(t)} v \frac{i(\varphi)}{\sigma_c} ds, \quad \forall v \in V, \quad (5.8)$$

where $\Gamma_p(t)$ is the boundary of the pit at time t. Here V is the trial space

$$V = \{v \in H^1(\Omega(t)) : v = 0 \text{ on } \Gamma_1\} \subset H^1(\Omega(t)),$$

where $H^1(\Omega(t))$ is a function space whose members, and their first derivatives, are square integrable (see, for example, [1] for details).

Suppose V_h is a finite dimensional subspace of V spanned by a collection basis functions. The FEM solution φ_h in $V_h \subset V$ is then found by solving

$$\int_{\Omega(t)} \nabla \varphi_h \cdot \nabla v_h d\Omega = \int_{\Gamma_p(t)} \frac{i(\varphi_h)}{\sigma_c} v_h ds, \quad \forall v_h \in V. \tag{5.9}$$

Let $\varphi_h \in V_h$ be given as the linear combination

$$\varphi_h = \sum_{j=1}^{N} \tilde{\varphi}_j \phi_j. \tag{5.10}$$

Choosing $v_h = \phi_k$, for $k = 1, 2, \ldots, N$, and using relation (5.10) gives

$$\sum_{j=1}^{N} \tilde{\varphi}_j \int_{\Omega} \nabla \phi_j . \nabla \phi_k d\Omega - \frac{1}{\sigma_c} \int_{\Gamma_p} i\Big(\sum_{j=1}^{N} \tilde{\varphi}_j \phi_j \Big) \phi_k ds = 0, \quad k = 1, 2, \ldots, N.$$

For each time, t, this is a system of non-linear equations which is solved using Newton's method.

MMPDELab requires the user to prescribe an appropriate mesh tensor, $\mathbb{M}_K$ (see (5.6)), to precisely control how the mesh automatically adapts to the changing solution and domain features. To ensure a sufficient number of elements in the evolving pit and around the heterogeneities we use a combination of mesh density functions. As in [25, 26] to resolve the pit boundary we use a modification of a distance-based monitor function given by

$$\mathbb{M}_K(x, y) = \left(1 + \frac{\mu_1}{\sqrt{\mu_2^2 d^2 + 1}} \right) I, \tag{5.11}$$

where

$$d(x, y) = \min_p |(x, y) - (x_p, y_p)|,$$

and (x_p, y_p) denotes any point on the boundary of the pit, Γ_{pit}, cf. [3]. This ensures that $\mathbb{M}_K$ will be largest in (x, y) regions where the distance to the pit boundary is the smallest; hence increasing the mesh resolution in these regions. The parameters μ_1 and μ_2 control the minimum mesh spacing and the rate at which mesh clustering occurs [22]. The effect of these parameters has been evaluated in [26].

To ensure additional resolution near material heterogeneities or inclusions we will also use the re-scaled version of Mackenzie's distance-based monitor function:

$$\tilde{\mathbb{M}}_K(x,y) = \begin{cases} 10 * \max(\mathbb{M}_K(x,y)) & \text{if } |\mathbb{M}_K(x,y) - (1+\mu_1)I| \leq 2.5 \text{ and } d_{ic}(x,y)) \leq 1 \\ \mathbb{M}_K(x,y) & \text{otherwise,} \end{cases} \tag{5.12}$$

where $d_{ic} = |(x,y) - (x_{ic}, y_{ic})|$, and $(x_{ic}$,$y_{ic})$ is the left or right corner of the inclusion. where the quantities (h,k) and r are chosen so that a circle with radius r and centered at (h,k) encloses the inclusion of interest. This monitor function is able to focus mesh points in the target region around the heterogeneities. Once the pit boundary encounters a material anomaly, two mesh solves are completed consecutively, first using the mesh density function (5.11) to resolve the pit boundary, and then using (**??**) to further refine the inclusion.

As discussed in [26], the initial mesh covering the initial pit geometry may be obtained with using a tool like **initmesh** in Matlab or **mesh node** in COMSOL, or indeed using the MMPDE above to ensure the initial pit geometry is resolved.

5.3.3 Mesh quality indicators

Throughout the simulation the mesh quality measure is monitored, and any elements with a mesh quality measure above a user prescribed tolerance are marked for refinement. These elements are then refined as described in the next section.

The equidistribution mesh quality measure of an element K used here is defined as

$$Q_{eq}(K) = \frac{N\mathbb{M}_K|K|}{\sigma_h},$$

where $|K|$ is the area of element K, N is the number of elements and

$$\sigma_h = \sum \int_K \mathbb{M}_K dx,$$

where the sum is taken over all elements in the mesh. Other mesh quality measures are available, see [12]. Q_{eq} is chosen as our mesh density function and is isotropic.

From the definition it is clear that $Q_{eq}(K) > 0$ for all elements K and

$$\frac{1}{N}\sum Q_{eq}(K) = 1.$$

Larger values of $\max_K Q_{eq}$ indicate that the mesh does not satisfactorily equidistribute $\mathbb{M}$.

5.3.4 A simple h-refinement strategy

Each element in the pit flagged as unsatisfactory as based on the mesh quality indicator is subdivided into 4 elements. This is accomplished by connecting the midpoints of each side of the triangular element, as shown in Figure 5.2.

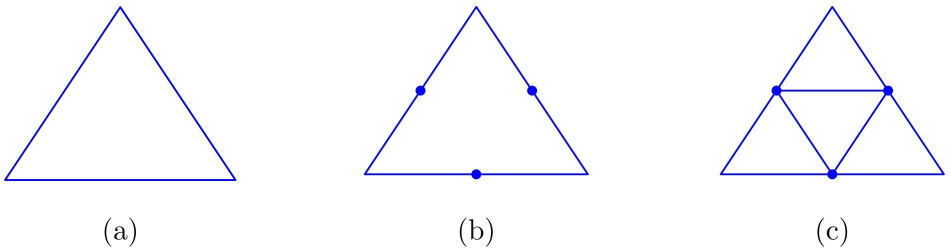
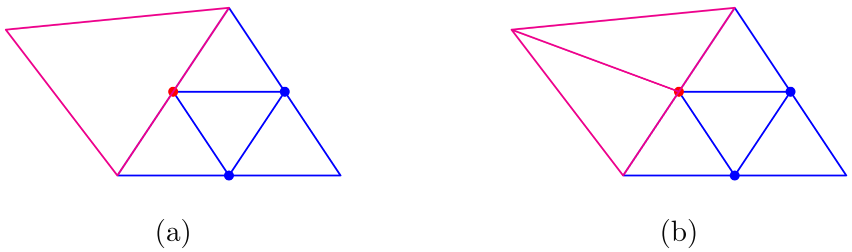


Figure 5.2: A simple h-refinement strategy: (a) mesh element to be refined, (b) add vertices on the midpoint of edges of the element, and (c) subdivide the element into 4 elements

Due to h-refinement, hanging nodes may appear on a refined element's edge and not coincide with nodes on the adjacent coarser element. For example, if one triangle is refined but not its neighbors. How to handle the hanging nodes in h-refinement strategy? We follow the steps below to fix the hanging nodes:

- Identify the hanging node(s) and neighboring coarse element (as shown in Figure 5.3(a)).



(a)

(b)

Figure 5.3: (a) Hanging node due to h-refinement, (b) Fixing the hanging node by subdividing the neighboring coarse element into 2 elements.

- Interpolate the values at the hanging nodes from the adjacent coarser element (as shown in Figure 5.3(b)).
- Finally update the element connectivity to include the newly created elements.

Once a h-refinement is complete, a mesh solve is performed to smooth the refined grid. Multiple mesh solves (or smoothing steps) may be completed to allow the now larger mesh to become a better approximation to the minimizer of the grid functional.

5.4 Numerical results

Table 5.2 illustrates the trends in the mesh quality measure, Q_{eq}, as the pit grows and interacts with a material anomaly when using r-refinement alone. Each cell in the table gives the number of elements on the pit and number of total elements which have a mesh quality measure above the specified tolerance as a pair. Generally, Table 5.2 shows that when using r-refinement on its own, the number of elements with a poor mesh quality measure increases as t increases, i.e., as the pit grows. Also as expected, the number of elements with poor mesh quality increases as the mesh quality tolerance decreases.

Table 5.2: The number of elements in the pit and on the whole domain with a Q_{eq} value larger than the tolerance for various times when using r-refinement alone.

Time (s) **Tolerance**	**50**	**51**	**70**	**71**	**90**	**91**
1.01	4 580	4 579	3 572	3 572	5 562	5 560
1.03	3 572	4 571	3 564	3 563	5 557	6 558
1.05	3 562	4 563	2 551	3 551	5 542	6 541
1.15	2 506	3 506	1 505	2 504	2 488	2 488
1.25	1 457	2 458	0 455	1 454	1 454	1 453
1.35	1 414	1 413	0 404	1 404	0 404	0 403
1.45	0 360	1 360	0 344	0 341	3 339	3 337
1.55	0 293	0 291	2 277	1 277	3 267	3 265
1.65	0 239	0 237	4 232	3 234	3 228	3 225
1.75	0 214	0 212	3 205	3 207	3 194	3 191

Table 5.3 indicates that completing h-refinements every 20 time steps is able to only slightly decrease the number of elements on the whole domain with poor grid quality. In fact, the number of poor quality elements in the pit itself actually increases with each h-refinement. This result is, in fact, not surprising. Dividing elements with poor mesh quality into 4 smaller elements does not generally improve the equidistribution of the mesh density function over the domain.

Table 5.3: The number of elements in the pit and on the whole domain with a Q_{eq} value larger than the tolerance for various times when using r-refinement and h-refinements every 20 s.

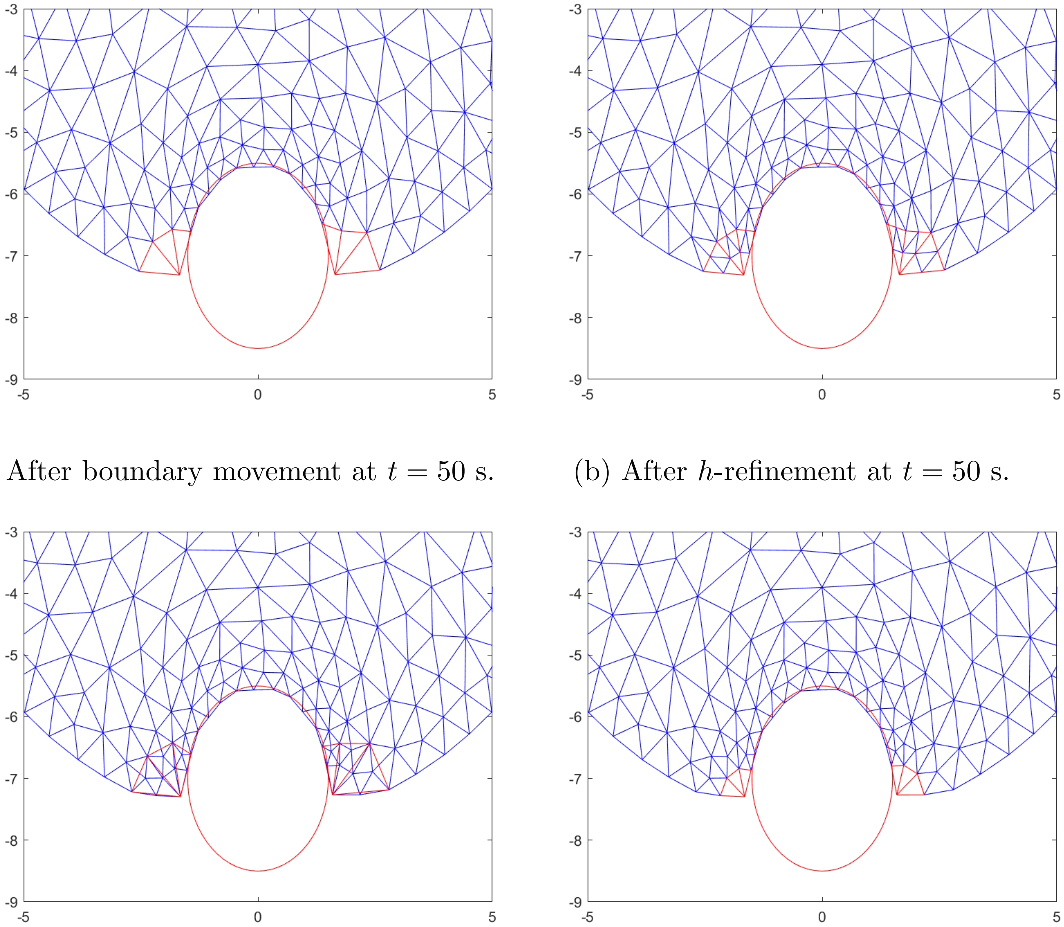
Time (s) **Tolerance**	**50**		**51**		**70**		**71**		**90**		**91**	
1.01	4	580	4	579	9	569	10	570	15	555	16	556
1.03	3	572	4	571	9	557	10	558	15	547	15	546
1.05	3	562	4	563	7	546	8	545	15	536	15	535
1.15	2	506	3	506	6	497	6	496	11	484	12	485
1.25	1	457	2	458	4	455	6	453	9	439	9	438
1.35	1	414	1	413	4	400	4	398	7	380	7	380
1.45	0	360	1	360	4	333	4	332	7	317	7	317
1.55	0	293	0	291	3	265	4	265	5	252	5	250
1.65	0	239	0	237	3	229	3	226	5	220	5	215
1.75	0	214	0	212	2	202	3	200	5	187	5	186

In Table 5.4, however, we see that if additional mesh PDE solves (or mesh smoothings) are performed then the additional elements provided by the h-refinement is ultimately able to improve the overall quality of the mesh.

Table 5.4: The number of elements in the pit and on the whole domain with a Q_{eq} value larger than the tolerance for various times when using r-refinement and h-refinements every 20 s followed by an additional 2 mesh smoothing steps.

Time (s) **Tolerance**	**50**	**51**	**70**	**71**	**90**	**91**
1.01	4 585	4 584	3 593	4 593	3 595	3 598
1.03	3 575	4 575	4 582	4 582	5 582	5 579
1.05	3 564	4 560	4 568	4 569	5 569	4 567
1.15	2 510	3 509	2 509	3 512	6 509	5 508
1.25	1 454	2 455	1 447	0 447	7 442	6 441
1.35	0 409	1 406	1 395	0 394	3 394	4 393
1.45	0 351	0 350	0 337	0 336	3 321	5 322
1.55	0 291	0 290	1 271	1 269	3 256	3 255
1.65	0 237	0 238	0 225	1 225	1 209	3 208
1.75	0 212	0 209	0 191	0 188	2 170	1 169

In Figure 5.4 we show a typical sequence in our hr-refinement approach. We simulate the growth of a corrosion pit as the boundary interacts with a corrosion resistant inclusion (outlined in red). In plot (a) the elements in the pit with a mesh quality measure larger than the tolerance 1.03 are highlighted in red. In plot (b) we show that each of these 6 red elements have now been refined once using the process described above. In plot (c) we show the mesh after one subsequent mesh solve or smoothing step. For reference, we have left the outline of the original red elements in place. After this single mesh solve, the mesh quality of each element can be computed again. We can see that we are still left with 6 red elements which are now much smaller than in plot (a).



(a) After boundary movement at $t = 50$ s.

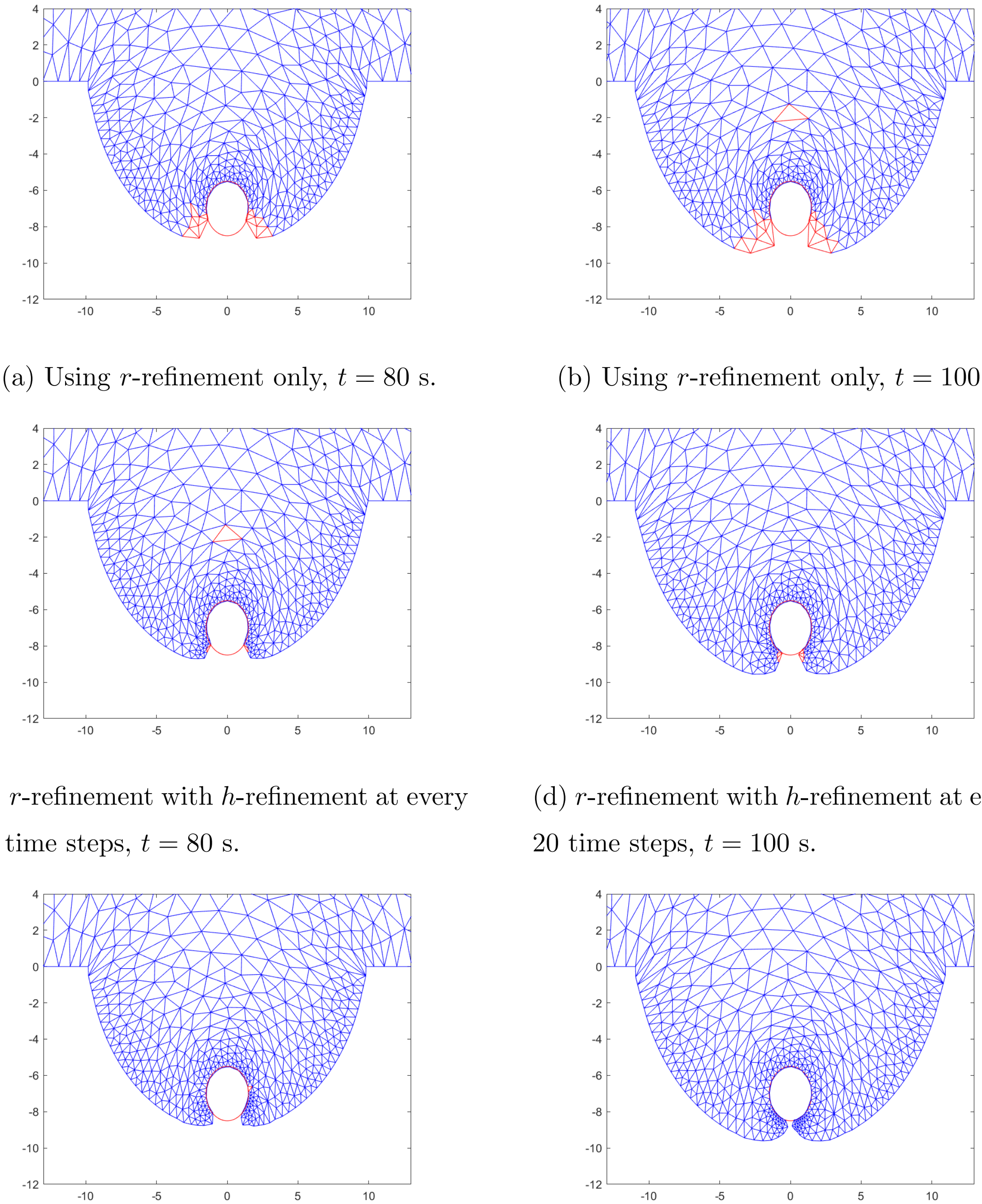
(b) After h-refinement at $t = 50$ s.

(c) After mesh solve at $t = 51$ s.

(d) After mesh solve at $t = 51$ s.

Figure 5.4: Identification of elements with poor mesh quality, h-refinement and subsequent smoothing for the simulation of a corrosion pit interacting with a corrosion resistant material.

In Figure 5.5 we continue a simulation of a corrosion pit interacting with a corrosion resistant material. We show 3 sets of results: using r-refinement alone (Figures (a) and (b)), h-refinement every 20 s (Figures (c) and (d)), and using h-refinement after r-refinement at every iteration (Figures (e) and (f)). Again mesh elements with poor mesh quality are highlighted in red. There are three obvious observations: (i) due to the increasing size of the pit and the fixed number of nodes used, the number



(a) Using r-refinement only, $t = 80$ s.

(b) Using r-refinement only, $t = 100$ s.

(c) r-refinement with h-refinement at every 20 time steps, $t = 80$ s.

(d) r-refinement with h-refinement at every 20 time steps, $t = 100$ s.

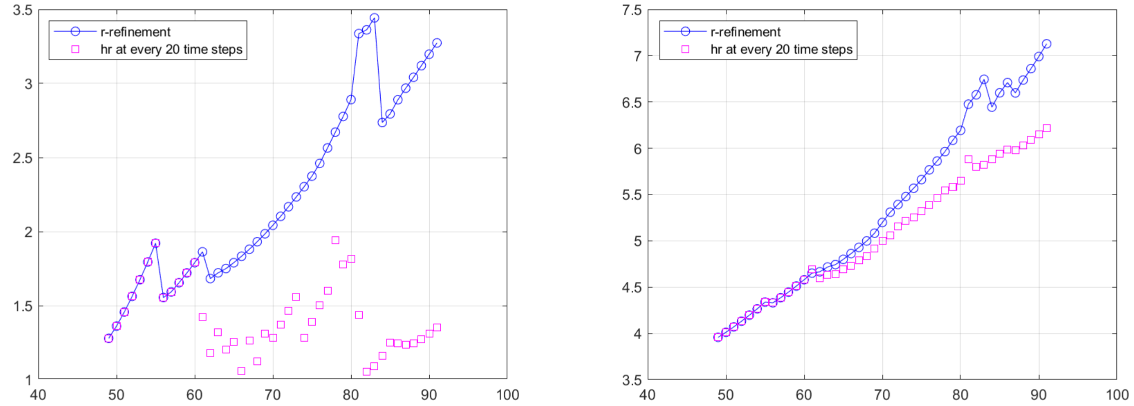
(e) r-refinement with h-refinement at every time step, $t = 80$ s.

(f) r-refinement with h-refinement at every time step, $t = 100$ s.

Figure 5.5: Meshes for a pit encountering a corrosion resistant inclusion at $t = 80$ and 100 s with r-refinement (top row), r-refinement with periodic h-refinement (middle row), and hr-refinement at every time step (bottom row).

of elements with poor mesh quality will generally increase with time if no h-refinement is used, (ii) periodic h-refinements can improve the resulting meshes — improving the resolution of the mesh in the targeted regions and in this simulation ultimately improving the overall mesh quality, and (iii) increasing the frequency of the h-refinement further improves the results, but of course at an increasing cost.

For the simulation above, in Figure 5.6 we plot the maximum and grid 2-norm of the mesh quality measure (in the corrosion pit) for the r-refinement (only) simulation and for the hybrid hr algorithm where h-refinement is completed at every 20 time steps. The maximum and grid 2-norm of the mesh quality increases with time, as the pit grows, when r-refinement is used without h-refinement. h refinement every 20 time steps reduces the norms of the mesh quality measure. In Figure 5.7, we plot

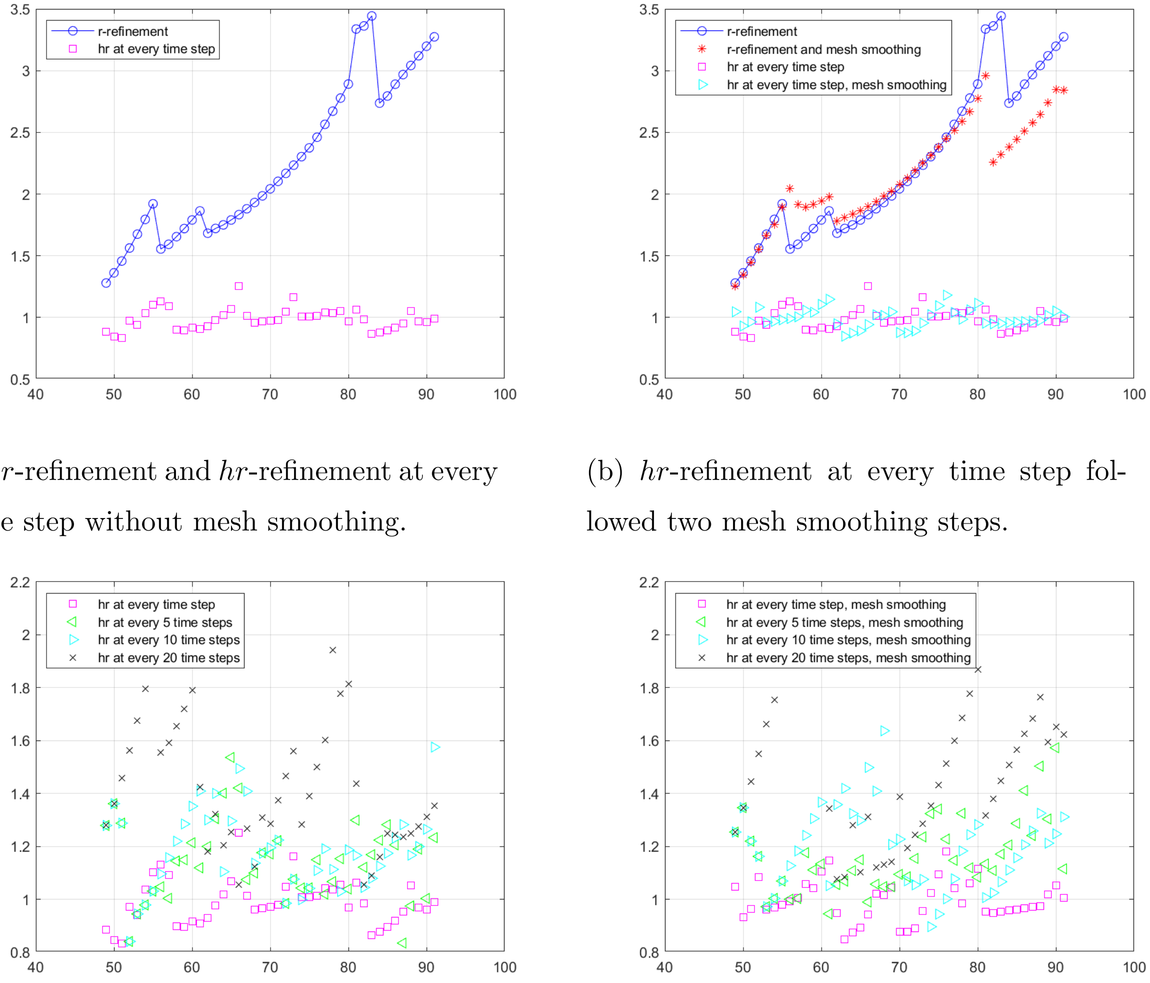


(a) Maximum norm.

(b) Grid 2-norm.

Figure 5.6: Norms of the mesh quality measure, Q_{eq}, computed in the pit for the (a) r-refinement and (b) hr-refinement algorithms (with h-refinement used every 20 time steps).

the maximum norm of the mesh quality measure (in the corrosion pit) for 4 variants of the hr approach: (i) r-refinement alone, (ii) r-refinement with two additional smoothing steps at each time step, (iii) hr-refinement at every time step, and (iv) hr-refinement at every time step followed by two additional smoothing steps. Generally the maximum norm of the grid quality measures increases if r-refinement is used alone with mesh smoothing steps improving the results some. Periodic h refinements help with results improving more significantly as the frequency of the h refinements and the number of smoothing steps increases.



(a) r-refinement and hr-refinement at every time step without mesh smoothing.

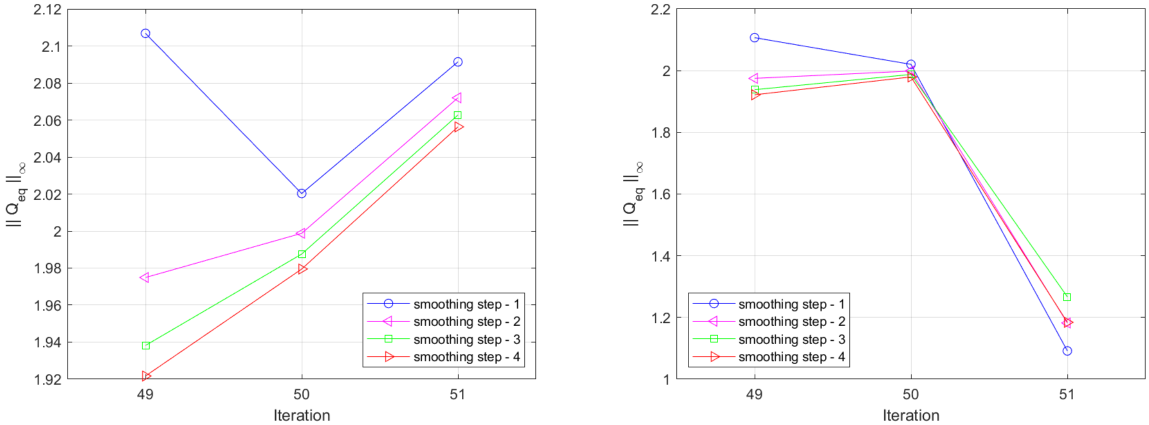
(b) hr-refinement at every time step followed two mesh smoothing steps.

(c) hr-refinement with various frequencies of h-refinements.

(d) hr-refinement followed by two smoothing steps with various frequencies of h-refinement.

Figure 5.7: The maximum norm of the mesh quality, Q_{eq}, on the pit as a function of time for r-refinement alone and various variants of hr-refinement.

In Figure 5.8 we see that the maximum norm of the mesh quality measure is improved by h-refinement at $t = 50$ s if followed by smoothing steps, and the result is further improved with each successive mesh smoothing step.

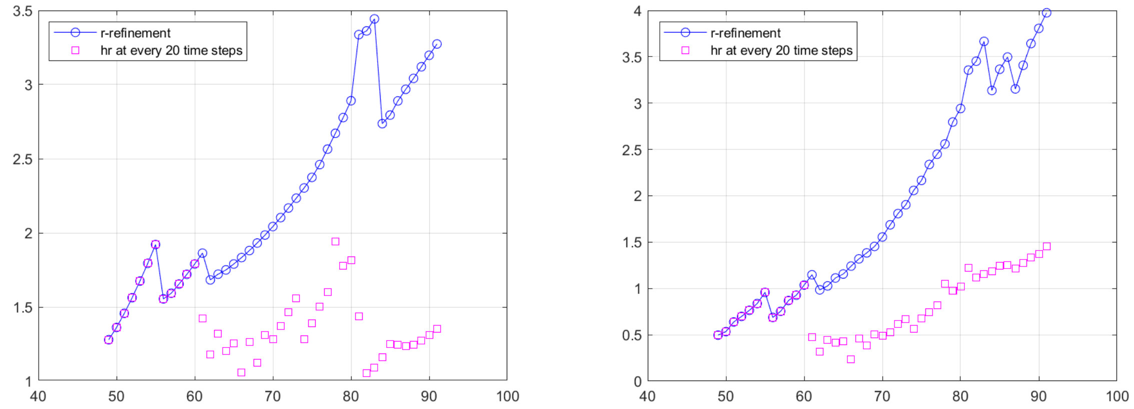


(a) Without h-refinement

(b) Using h-refinement at $t = 50$ s

Figure 5.8: The impact of the number of smoothing steps on the maximum norm of the mesh quality, Q_{eq}, with (a) and without (b) h-refinement at $t = 50$ s.

In Figure 5.9 we see that if we compute the norms of the mesh quality measure just over the elements whose mesh quality measure exceeds the mesh quality tolerance of 1.03, then both the maximum and grid 2-norms are greatly reduced with periodic h-refinements. Above, we saw that the grid 2-norm of the mesh quality measure over the pit was reduced, but only slightly so, by the h-refinement. This is sensible since the grid 2-norm over the whole pit is influenced greatly by the large number of elements with mesh qualities below the mesh quality tolerance.

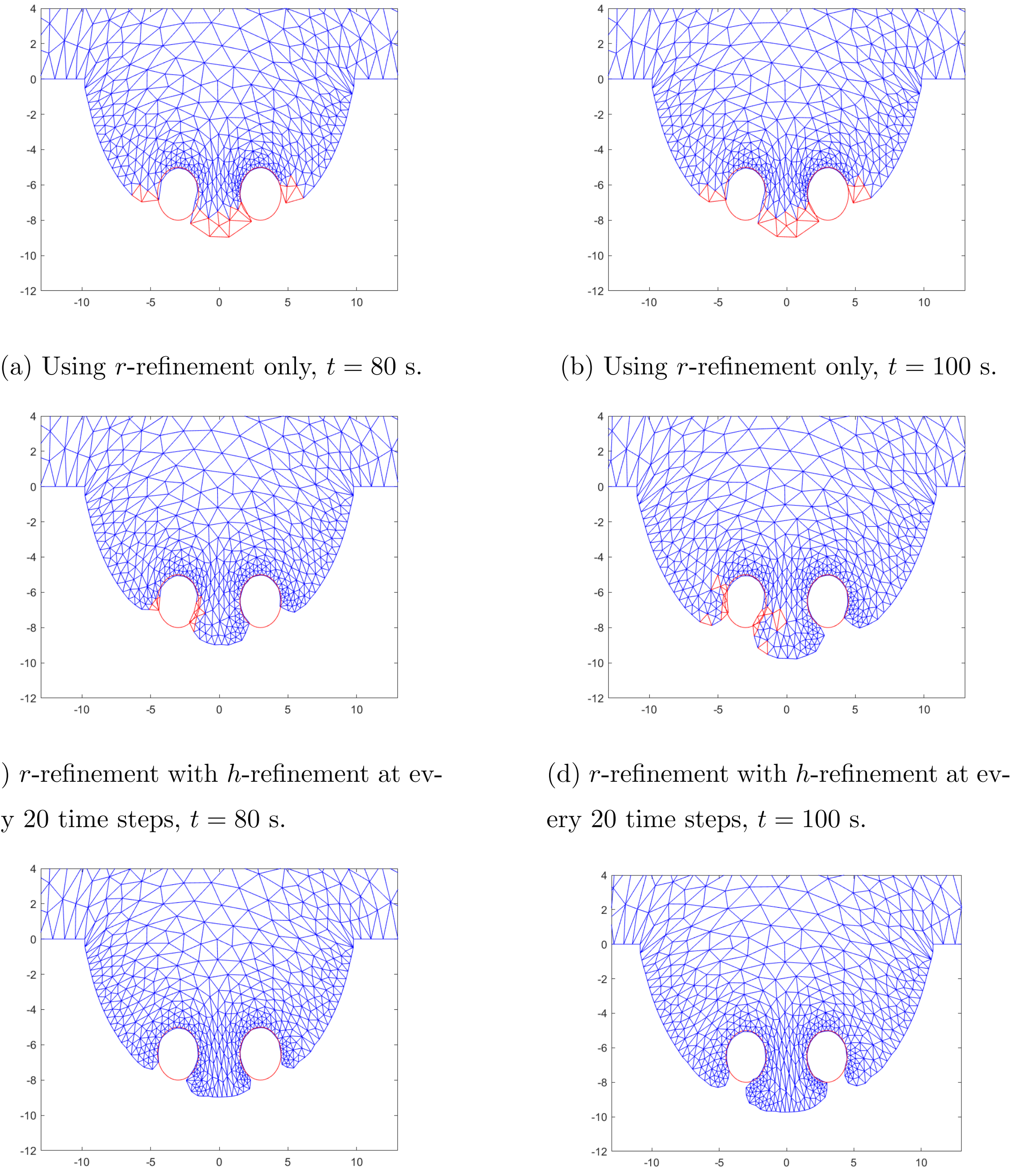


(a) Maximum norm

(b) Grid norm

Figure 5.9: Mesh quality Q_{eq} for the bad elements on the pit, hr-refinement is used at every 20 iterations.

As we saw in [25], designing mesh density functions capable of maintaining sufficient mesh resolution and quality in the presence of multiple heterogeneities is rather difficult. Indeed, this was one of the primary reasons to propose an hr-refinement strategy. In Figure 5.10 we illustrate the simulation of pit growth as the pit encounters two corrosion resistant inclusions. We see that r-refinement alone has difficulty with the competing features of the solution, and there is a loss of mesh quality around, and in between, the corrosion resistant anomalies. Following the r-refinement with h-refinements every 20 time steps improves the results. As the frequency of h-refinements is increased (in tandem with the r-refinement) we recover a high quality resolution of the pit geometry as the pit grows around the inclusions.



(a) Using r-refinement only, $t = 80$ s.

(b) Using r-refinement only, $t = 100$ s.

(c) r-refinement with h-refinement at every 20 time steps, $t = 80$ s.

(d) r-refinement with h-refinement at every 20 time steps, $t = 100$ s.

(e) r-refinement with h-refinement at every time step, $t = 80$ s.

(f) r-refinement with h-refinement at every time step, $t = 100$ s.

Figure 5.10: Meshes for a pit encountering two corrosion resistant inclusions at $t = 80$ and 100 s with r-refinement (top row), r-refinement with periodic h-refinement (middle row), and hr-refinement at every time step (bottom row).

5.5 Conclusion

This paper presents a robust, fully automatic, moving mesh solution framework for pitting corrosion in heterogeneous materials or situations where material voids are present. The moving mesh approach is able to continuously and smoothly evolve a fixed mesh topology according to the changing pit geometry as the pits encounter inclusions with varying crystallography or voids.

Designing mesh density functions capable of resolving corrosion dynamics in situations with several competing regions of interest in difficult and would certainly benefit from an *hr*–refinement strategy (which both redistributes nodes as we have presented here but also allows periodic changes to the number of mesh nodes). This work is ongoing and will appear in a subsequent paper.

5.6 Data availability

The raw or processed data required to reproduce these figures and findings cannot be shared at this time due to technical or time limitations, but are available from the authors upon request.

Acknowledgements

The work of Ronald D. Haynes and Abu N. Sarker (PhD student) has been supported by a NSERC discovery grant (Canada).

Bibliography

[1] David R Adams and Lars I Hedberg. *Function Spaces and Potential Theory*, volume 314. Springer Science & Business Media, 1999.

[2] Allen J Bard, Larry R Faulkner, and Henry S White. *Electrochemical Methods: Fundamentals and Applications*. John Wiley & Sons, 2022.

[3] G Beckett, John A Mackenzie, and ML Robertson. A moving mesh finite element method for the solution of two-dimensional Stefan problems. *Journal of Computational Physics*, 168(2):500–518, 2001.

[4] Dennj De Meo and Erkan Oterkus. Finite element implementation of a peridynamic pitting corrosion damage model. *Ocean Engineering*, 135:76–83, 2017.

[5] Virginia G DeGiorgi, Nithyanand Kota, Alexis C Lewis, and Siddiq M Qidwai. Numerical modeling of pit growth in microstructure. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55850, pages 1–7. American Society of Mechanical Engineers, 2013.

[6] Ravindra Duddu. Numerical modeling of corrosion pit propagation using the combined extended finite element and level set method. *Computational Mechanics*, 54(3):613–627, 2014.

[7] MA Hill, JF Bingert, and RS Lillard. The relationship between crystallographic orientation and the passivity and breakdown of beryllium. In *194th Meeting of the Electrochemical Society*. 1998.

[8] CW Hirt, AA Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.

[9] Cyrill W Hirt, Anthony A Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[10] Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *Journal of Computational Physics*, 174(2):903–924, 2001.

[11] Weizhang Huang. An introduction to MMPDElab. *ArXiv Preprint ArXiv:1904.05535*, 2019.

[12] Weizhang Huang and Robert D Russell. *Adaptive Moving Mesh Methods*, volume 174. Springer Science & Business Media, 2010.

[13] Siavash Jafarzadeh, Ziguang Chen, and Florin Bobaru. Computational modeling of pitting corrosion. *Corrosion Reviews*, 37(5):419–439, 2019.

[14] Hormoz Jahandari, Scott MacLachlan, Ronald D. Haynes, and Niall Madden. Finite element modelling of geophysical electromagnetic data with goal-oriented *hr*-adaptivity. *Computational Geosciences*, 24(3):1257–1283, 2020.

[15] SMLC Jain, MLC Lim, JL Hudson, and JR Scully. Spreading of intergranular corrosion on the surface of sensitized Al-4.4 mg alloys: A general finding. *Corrosion Science*, 59:136–147, 2012.

[16] Patrick M Knupp, Len G Margolin, and Mikhail Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176(1):93–128, 2002.

[17] EV Koroleva, GE Thompson, P Skeldon, and B Noble. Crystallographic dissolution of high purity aluminium. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 463, pages 1729–1748. The Royal Society, 2007.

[18] Nithyanand Kota, Siddiq M Qidwai, Alexis C Lewis, and Virginia G DeGiorgi. Microstructure-based numerical modeling of pitting corrosion in 316 stainless steel. *ECS Transactions*, 50(31):155–164, 2013.

[19] D Krouse, N Laycock, and C Padovani. Modelling pitting corrosion of stainless steel in atmospheric exposures to chloride containing environments. *Corrosion Engineering, Science and Technology*, 49(6):521–528, 2014.

[20] B Ravi Kumar, Raghuvir Singh, Bhupeshwar Mahato, PK De, NR Bandyopadhyay, and DK Bhattacharya. Effect of texture on corrosion behavior of aisi 304l stainless steel. *Materials Characterization*, 54(2):141–147, 02 2005.

[21] Nicholas J Laycock, Donal P Krouse, Shaun C Hendy, and David E Williams. Computer simulation of pitting corrosion of stainless steels. *The Electrochemical Society Interface*, 23(4):65–71, 2014.

[22] JA Mackenzie and ML Robertson. The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method. *Journal of Computational Physics*, 161(2):537–557, 2000.

[23] Len G Margolin. Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):198–202, 1997.

[24] Fabio Nobile and Luca Formaggia. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-West Journal of Numerical Mathematics*, 7:105–132, 1999.

[25] Abu N. Sarker, Ronald D. Haynes, and Michael Robertson. A moving mesh method for pitting corrosion of heterogeneous materials, 2023. Preprint.

[26] Abu N. Sarker, Ronald D. Haynes, and Michael Robertson. Moving mesh simulations of pitting corrosion, 2023. Preprint, Submitted.

[27] Akinori Sato, Kenzo Kon, Shigeo Tsujikawa, and Yoshihiro Hisamatsu. Effect of crystallographic orientation on dissolution behavior of stainless steels single crystal. *Materials Transactions, JIM*, 37(4):729–732, 1996.

[28] Stefan Scheiner and Christian Hellmich. Stable pitting corrosion of stainless steel as diffusion-controlled dissolution process with a sharp moving electrode boundary. *Corrosion Science*, 49(2):319–346, 2007.

[29] Stefan Scheiner and Christian Hellmich. Finite volume model for diffusion-and activation-controlled pitting corrosion of stainless steel. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2898–2910, 2009.

[30] Jong Hyun Seo, Jong-Ho Ryu, and Dong Nyung Lee. Formation of crystallographic etch pits during AC etching of aluminum. *Journal of the Electrochemical Society*, 150(9):B433–B438, 2003.

[31] SM Sharland. A review of the theoretical modelling of crevice and pitting corrosion. *Corrosion Science*, 27(3):289–323, 1987.

[32] SM Sharland. A mathematical model of crevice and pitting corrosion—II. the mathematical solution. *Corrosion Science*, 28(6):621–630, 1988.

[33] SM Sharland and PW Tasker. A mathematical model of crevice and pitting corrosion-I. The physical model. *Corrosion Science*, 28(6):603–620, 1988.

[34] Laura B Simon, Mohammad Khobaib, Theodore E Matikas, CS Jeffcoate, and MS Donley. Influence of pitting corrosion on structural integrity of aluminum alloys. In *Nondestructive Evaluation of Aging Materials and Composites III*, volume 3585, pages 40–47. International Society for Optics and Photonics, 1999.

[35] Grainne M Treacy and Carmel B Breslin. Electrochemical studies on single-crystal aluminium surfaces. *Electrochimica Acta*, 43(12):1715–1720, 1998.

[36] A Turnbull, DA Horner, and BJ Connolly. Challenges in modelling the evolution of stress corrosion cracks from pits. *Engineering Fracture Mechanics*, 76(5):633–640, 2009.

[37] John C Walton. Mathematical modeling of mass transport and chemical reaction in crevice and pitting corrosion. *Corrosion Science*, 30(8-9):915–928, 1990.

[38] Kai Wang, Chenpei Li, Yanhui Li, Jinling Lu, Yueshe Wang, and Xingqi Luo. Multi-physics coupling analysis on the time-dependent localized corrosion behavior of carbon steel in CO_2-H_2O environment. *Journal of The Electrochemical Society*, 167(1):013505–013505, 2019.

[39] BV Wells, Michael J Baines, and Paul Glaister. Generation of arbitrary Lagrangian–Eulerian (ALE) velocities, based on monitor functions, for the solu-

tion of compressible fluid equations. *International Journal for Numerical Methods in Fluids*, 47(10-11):1375–1381, 2005.

[40] M Yasuda, F Weinberg, and D Tromans. Pitting corrosion of Al and Al-Cu single crystals. *Journal of the Electrochemical Society*, 137(12):3708–3715, 1990.

[41] Xingliang Yu, Fangyi Wan, and Yingnan Guo. Micromechanics modeling of skin panel with pitting corrosion for aircraft structural health monitoring. In *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE, 2016.

Chapter 6

Conclusion and future work

In this chapter, we present a summary, the key contributions of this thesis, and recommendations for future work. Our research focuses on the areas of adaptive moving meshes, pitting corrosion, and finite element methods. Specifically, we have contributed to the development of software for adaptive algorithms in complex domains, the study of heterogeneous materials with an emphasis on inclusion-type domains, and the r and hr-refinement methodology, which enhances the accuracy and efficiency of numerical simulations of moving boundary problems. Our recommendations for future work include further studies on moving boundary problems, numerical analysis of the methods, corrosion simulations for different metals, and coupling our approach with domain decomposition strategies.

6.1 Summary of the thesis and our contribution

This thesis presents a proof of concept r and hr refinement approach to solving the moving boundary problem related to pitting corrosion. Firstly, we demonstrated the effectiveness of the r refinement approach in handling moving boundaries and changing topology due to merging pits in pitting corrosion problems. Secondly, we conducted numerical simulations of pitting corrosion in heterogeneous materials, which is challenging due to the presence of inclusion-type domains. We have shown that r refinement could provide quality meshes near the inclusion and pit as required. Finally, we have demonstrated that hr-refinement can significantly improve the resolution near the features of interest.

An adaptive moving mesh method is a powerful technique for obtaining a more accurate solution. This thesis has reviewed two types of adaptive moving mesh techniques: r-refinement and h-refinement. In **Chapter 1**, we discussed the objectives and scope of the thesis, as well as relevant literature on moving mesh and moving boundary problems, including pitting corrosion. In **Chapter 2**, we provided background materials for this thesis, including an overview of the moving mesh method, the formulation of MMPDEs, mesh adaptation functions, and solution techniques for the physical PDE and mesh equation. There are two ways to solve the physical PDE and mesh equation: simultaneously and alternately. However, the simultaneous approach has a highly nonlinear coupling between the physical PDE and mesh PDE. Therefore, we have chosen the simplest approach to the moving boundary problem and then decoupled the boundary movement from the mesh generation. We have generated the mesh and solution using the alternating approach. In addition, we have presented the preliminaries of the mechanism of pitting corrosion, crystal orientation and a model problem.

In **Chapter 3** we have shown how to design and implement an adaptive moving mesh method for a moving boundary problem related to pitting corrosion with single and multiple pits. The adaptive mesh is generated automatically by solving a mesh PDE coupled to the nonlinear potential problem. We have shown the moving mesh approach enables initial mesh generation, provides mesh recovery, and is able to tackle changing pit geometry smoothly. To do this, we designed a metric for mesh adaptation based on geometric considerations and on the solution of the PDEs, indicating where the mesh needs to be moved to capture fine-scale features or moving boundaries. In addition, we have introduced a technique to move the corner nodes of the pits. Single and multiple pits are considered, as are materials with different crystal direction(s). A procedure has been presented which allows pits to merge without a change in mesh topology, allowing computation to proceed without completely restarting the computation.

Chapter 4 is devoted to demonstrating an adaptive moving mesh method for simulating pitting corrosion in materials containing heterogeneous inclusions. Inclusions are regions of a material that have a different composition or properties than the surrounding material. In this chapter, we discussed how to handle multiple inclusions and voids in the computational domain. We compute a metric for mesh adaptation based on the solution of PDEs, indicating where the mesh needs to be moved to capture the moving boundary and the metric was modified to handle inclusions. We have shown that r-refinement approach is capable of handling changing pit geometry and materials with varying crystallography and corrosion-resistant inclusions.

We observed that in some cases, r-refinement alone could not provide high mesh density near the inclusion(s) for a long simulation time due to the obstacle(s) and the moving front. We proposed a combination of h- and r- refinement to overcome this. h-refinement adds mesh elements by dividing each existing element into two or more elements and maintaining the type of element used.

In **Chapter 5**, we designed and implemented an adaptive hr-refinement procedure for the simulation of pitting corrosion with heterogeneous materials. The adaptive hr-refinement was able to handle changing pit geometry, including materials with varying crystallography and corrosion-resistant inclusions. We have shown that hr-refinement can significantly improve the mesh quality of the simulation compared to r-refinement alone.

In summary, we have presented a robust, fully automatic, moving mesh solution framework for pitting corrosion in materials with homogeneous and heterogeneous with and without inclusions. The research also focused on developing software, a new extension of MMPDElab providing hr–refinement.

6.2 Recommendations for future work

Future research primarily falls under the umbrellas of adaptive moving meshes, corrosion processes, and domain decomposition methods. This involves the development of adaptive algorithms in complex domains, the further study of heterogeneous materials with an emphasis on inclusion-type domains of an arbitrary number, and improvements to the *hr*-refinement methodology. Since this thesis work is interdisciplinary, there are various areas where future work can be conducted and some recommendations for future work are given below.

1. **Numerical simulation of pitting corrosion for many pits:** We presented a proof of concept r and hr refinement approach to the moving boundary problem related to pitting corrosion. The simulation of large pit growth or the initiation of many pits would likely benefit from an *hr*–refinement strategy which both redistributes nodes as we have presented here but also allows periodic changes to the number of mesh nodes.

2. **Coupling the boundary movement with the mesh movement:** In this thesis work, the boundary movement and mesh movement are handled separately. Coupling the boundary movement implicitly with the mesh movement would also be a subject of future work. This will require further modification of the MMPDElab software.

3. **Improvements to the model by including more physics:** In the existing model we can include more physics, which will modify the physical PDE to more accurately model the physical problem.

4. **Simultaneous approach for solving the pitting corrosion PDE model and the mesh movement:** As we discussed, to solve the physical PDE and mesh equation, we may solve the physical and mesh PDEs in a simultaneous or alternating fashion. We have taken the simplest (alternating) approach for solving the moving boundary problem and have decoupled the boundary movement from the mesh generation. In future work, the simultaneous solution approach will be explored. The main disadvantage of this approach is that it has highly nonlinear coupling between the physical solution and the mesh. The advantage is the reduction in the time lag between the solution and the mesh.

5. **Error based hr-refinement methods for pitting corrosion:** The MM-PDE software only provides a P1 finite element of the solution. Extending the software to give both a P1 and P2 finite element solution would allow the computation of an error estimate. This error estimate could then be used as part of the mesh density function and as a trigger for h-refinement.

6. **Domain decomposition simulation for pitting corrosion:** Domain decomposition methods are widely used in scientific computing community to solve the problem efficiently and quickly. The simulation of large pit growth or the initiation of many pits would likely benefit from an hr–refinement strategy coupled with a domain decomposition approach to allow the problem to be spatially partitioned and the computation distributed to harness additional processors.

7. **Parallel domain decomposition approach for simulating pitting corrosion PDE model in 3D:** Other future work could include looking at pitting corrosion in 3D on general surfaces. Recently, a closest-point method library for solving PDEs on surfaces with parallel domain decomposition was introduced by May et al. [1]. In particular, the library is able to solve elliptic and parabolic equations including reaction-diffusion equations. This parallel domain decomposition technique can be used for simulation of pitting corrosion PDE model in 3D.

In conclusion, moving mesh approaches show a lot of promise for dealing with moving boundary problems, especially when working with complex domains with inclusions. Although, moving mesh techniques have been employed in a variety areas, the field will benefit from further uptake in application areas. The MMPDElab package should be made accessible in a number of programming languages and integrated with other software applications, such as the PETSc (Portable, Extensible Toolkit for Scientific Computation) library, for the benefit of the scientific computing community and numerical analysts.

,

Bibliography

[1] Ian CT May, Ronald D Haynes, and Steven J Ruuth. A closest point method library for PDEs on surfaces with parallel domain decomposition solvers and preconditioners. *Numerical Algorithms*, pages 1–23, 2022.