

Interim Report: Hybrid Methods for Finite Element Meshing

Jack Bradbrook (psyjb4)

November 30, 2016

Contents

1	Introduction	3
2	Motivation and Background	3
3	Related Work	4
3.1	Traditional Stress refinement methods	4
3.2	Uses of Artificial Intelligence and Machine Learning	5
3.3	Quality metrics	6
4	Description of the work	7
4.1	Aims and Objectives	7
5	System specification	7
6	Functional Requirements	7
7	Non-Functional Requirements	8
8	System Design	9
8.1	System Overview	9
8.2	Modular Architecture	9
8.3	Input Files	9
8.4	Multi Threading	9
9	Software Implementation	9
9.1	Third Party FE Application	9
9.2	LISA	10
9.3	Languages and platforms	11
9.4	Implementation Methodology	11
9.5	Implementation of Subsystems	11
9.5.1	Re-meshing using hierarchical refinement	11
9.5.2	Stress Based Refinement	12
9.5.3	Rule Based Refinement	12
9.5.4	Interfacing with LISA and the FE data model	12
9.5.5	Mesh Quality Assessment	14
9.6	Modelling Real World problems	14
9.6.1	Suspension Bridge structure	14
9.6.2	Validating system against previously obtained research results	14
9.7	Implementation Issues	15
9.7.1	Sorting Element Nodes in 3d space using convex hull Algorithm	15

9.7.2	Attempts to Automatically Define edges within models	17
9.7.3	18
10	Evaluation of Project	18
11	Functional testing	18
12	Non functional testing	18
12.1	Increase in performance through parallel execution	18
12.2	Quality	18
12.3	Documentation	18
12.4	Maintainability	19
12.4.1	Evaluation Issues	19
12.5	Evaluation of overall project	19
12.6	Evaluation summary	19
12.7	Successes and Limitations	19
13	Further Work	19
13.1	Gathering feedback from experienced engineers	19
13.2	Improving usability through a web interface	19
14	Personal Reflections and Summary	20
	References	22

1 Introduction

The overall aim of this project is to perform the task of refining a Finite Element Mesh (FEM) using a stress based method as is typical of FEM refinement in conjunction with an approach that uses techniques from Artificial Intelligence and Machine Learning.

Finite Element Analysis (FEA) is a method widely used across different engineering domains to simulate structures under certain conditions. The method works by taking a geometry defined in continuous terms, discretising it into a mesh system before calculating the property values for each of the discrete elements. This allows engineers to observe the effect the conditions have on the entire structure, see figure 1.

Success of the project will be determined by implementation of a system that is able to combine the two approaches described above in order to refine a mesh to of a quality comparable to that of a successful stress based method.

2 Motivation and Background

Over the past forty years FEA has emerged as a prominent technology for simulating complex real world engineering problems [1, 5]. FEA works by solving a system of differential equations with each equation representing a single element in a geometric mesh. By doing this FEA is able to generate highly accurate approximations for the properties of complex physical systems [5] [18]. The method can also be highly computationally expensive with the complexity typically increasing exponentially with the model size [5]. Analysis therefore proves to be highly time consuming and costly for individuals and organisations to conduct [7]. [16]

Finite Element Models have several inherent properties which for the most part specified by the engineers who create them, the settings for these properties greatly determine the results obtained from the models execution. The first of these properties of an FE model begin with there is the mesh which is constructed out of nodes (points which are intersections of edges that form elements) as well as the elements themselves, either a polygon or a polyhedron between the nodes. Nodes and elements are important concepts as they provide the theoretical framework for reasoning about the other properties of a mesh and hence the overall quality of the model [18].

elements represent

The essence of

An important consideration when conducting FEA is the trade-off of a models accuracy against the time in which it can be solved. A major variable determining this trade-off is the models mesh structure which discretizes the problem so that simulation can be run on it. A mesh that is finer is more computationally expensive but also produces results of greater accuracy. It is therefore desirable to generate a mesh which is fine where accuracy is most needed but coarse where it isn't [17].

In every type of analysis that the FE method is used for (thermal, structural, fluid flow, electrical) there is a specific differential equation associated with each of the elements. In order to achieve overall convergence of the model the equations must be solved simultaneously to achieve a value for each of the discrete elements [18]. For this dissertation project attention will be given specifically to the problem of FEA meshing in the context of static structural analysis where the value calculated across each element is its stress. It makes sense to work on hybrid mesh refinement in the context of static structural analysis as it is perhaps the most common engineering application of the method and as such has the largest body of research that is relevant [5][18].

For engineers the value obtained through computing the stresses under a particular set of conditions is feedback on the quality of their design. Ideally the results from an analysis will provide a good understanding of where the design is weak and how concentrated this weakness is. This information is used to either help verify the designs' quality or alternatively inform changes to its geometry or material properties so as to

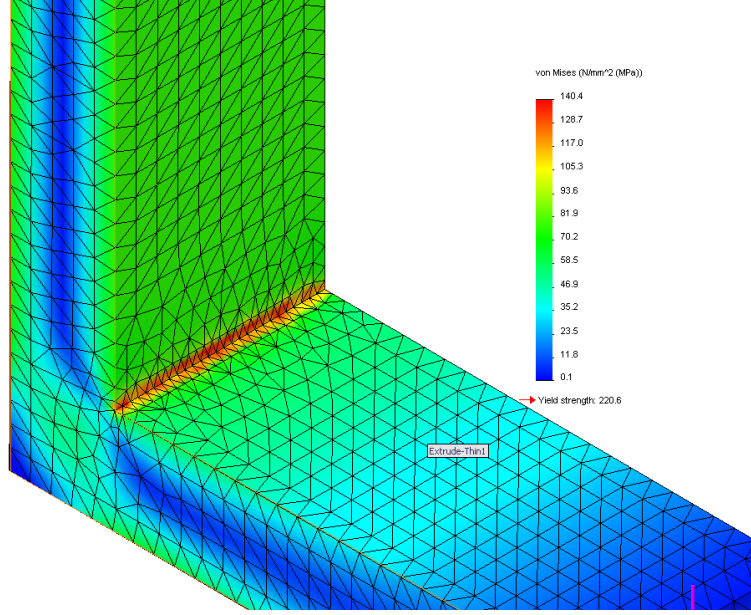


Figure 1: Mesh refinement in corner under high stress image source: ([27])

reduce stress on subsequent analysis [20].

To understand the gradient of stress within a part of the model the mesh needs to be designed carefully. As each element can only display values calculated from its edge nodes a smooth gradient requires a higher concentration of elements in areas under higher stress. A high quality mesh is therefore considered to have a higher concentration of elements in areas of predicted high stress while retaining lower concentration elsewhere, thereby revealing weaknesses in the design while minimising the models runtime.

Traditionally the automated mesh refinement process consists of computing stresses for a model with an initial coarse mesh and low computation cost, once rough stresses have been computed the elements in areas of higher stress can be divided recursively into additional elements in order to achieve smoother gradients on further executions [16]. Figure 1 shows a mesh which has been refined in an area of higher stress thus providing a clearer indication of a components weakness.

Unfortunately for many large models this method for refining a mesh is still excessively costly [4]. It therefore seems worth investigating use of alternative approaches posed by the field of computer science and artificial intelligence that could support the traditional high stress meshing approach.

3 Related Work

Many approaches have so far been taken in an attempt to improve a computer’s ability to perform the task of finite element meshing. The following subsections present an overview of the research conducted for the various aspects of the project which are more explicitly outlined under “Aims and Objectives”

3.1 Traditional Stress refinement methods

A number of methods exist to perform mesh refinement based on a stress gradient for a mesh, with the most common being higherarchical refinementent also known as h-refinement and relocation refinement or

r-refinement[2] [3]

h-refinement: H-refinement is the process of recursively refining a mesh by splitting elements into additional sub elements. This process can be performed for elements with both both triangular and quadrilateral shapes. [2].

r-refinement: R-refinement is a method which attempts to improve the quality of the stress gradient without the alteration of the mesh element count and thus the computational cost. This is achieved by the relocation of elements within the mesh which effectively increases the size of elements in areas of low stress, while reducing the size in areas where stress is high [3].

3.2 Uses of Artificial Intelligence and Machine Learning

Within the domains of AI and machine learning methods such as neural networks [9], case based reasoning [10] and inductive logic programming [5] have all been adopted to facilitate generation of meshes comparable to that of human experts. Similarly there has also been effort to combine multiple numerical methods simultaneously for solving re meshing problems [8] although effort to combine the two approaches does not appear to have so far been made.

Due to the difficulty of obtaining meshes which hold commercial interest the majority of researchers working in this field have had to result to the use of training sets developed within academia [4]. The primary issue associated with this is that often the training data does not exhibit the level of complexity that you would expect in many industrial sectors. Many researchers must accept this as a limitation or agree commercial terms with an organisation in order to gain access to their models [14].

Having reviewed a variety of different AI based applications to FE the use of Inductive Logic Programming (ILP) used by Bojan Dolsak et al is of greatest interest. ILP is a machine learning method first presented by Stephen Muggleton in his 1991 paper "Inductive Logic Programming" [11]. Muggleton suggests that the traditional approaches of machine learning which rely on use of extensive data sets and statistical analysis are poor in the case of many real world problems for which data is not available [13]. Muggleton cites Human learning as an example of use of ILP style techniques where understanding of new concepts is achieved not through crunching large volumes of data points but instead the use of induction on a relatively concise set of background facts and examples obtained from previous life experiences [13].

ILP uses three types of input information in order to hypothesise additional facts about the system. These three types of input information are:

- Positive examples of what constitutes an area that is well meshed
- Negative example of areas that are poorly meshed
- Background facts

Using this information ILP is capable of hypothesising rules by determining which rules can exist within the system where given the set of background facts all positive examples are satisfied while few or none of the negative examples are. Although ILP requires a body of additional metadata associated with each mesh this is easier to obtain making ILP a highly practical solution. Along with his publication Muggletons also released his implementation of an ILP algorithm as a program titled "Golem" [12], Golem was applied by Dolsak to the problem of mesh refinement with a training set of just five meshes [5]. The resulting rule set when applied to subsequent models was able to correctly classify and re mesh areas with an average accuracy of 78% for a range of geometries [5] [6].

Dolsak’s choice of metadata for the ILP method to generate mesh rules is based on the classification of edges within the FE model which he knew to be crucial to determining their strength. For example if it is know that an edge within the model has a force applied close to it based on the initial model conditions then other edges that intersect it should be meshed to have a greater concerntration of adjacent elements prior to model execution. [4] [6].

The format of the rules make them attractive for experimenting with as part of a hybrid method since the method determines how to refine the mesh based on the arrangement of edges. This detail of analysis of the mesh is at a comparable detail to that of a traditional splitting method such as h-refinement which will likely to improve the ease at which the two methods can be combined simultaneously in the latter stages of the project.

3.3 Quality metrics

Finally work has also been done on establishing valid metrics for assessing the quality of a mesh automatically [14, 9] Metrics for meshes have been research far more extensively that AI methods due to their use for comparing different stress based refinements. There are also cases of common metrics being used for industrial meshing applications [14]. Although there are metrics for assessing a mesh on a global level such as element count score [14] the consensus is that due to the variation in meshes this is less reliable than assessing quality based on the properties of individual elements within the mesh [14]

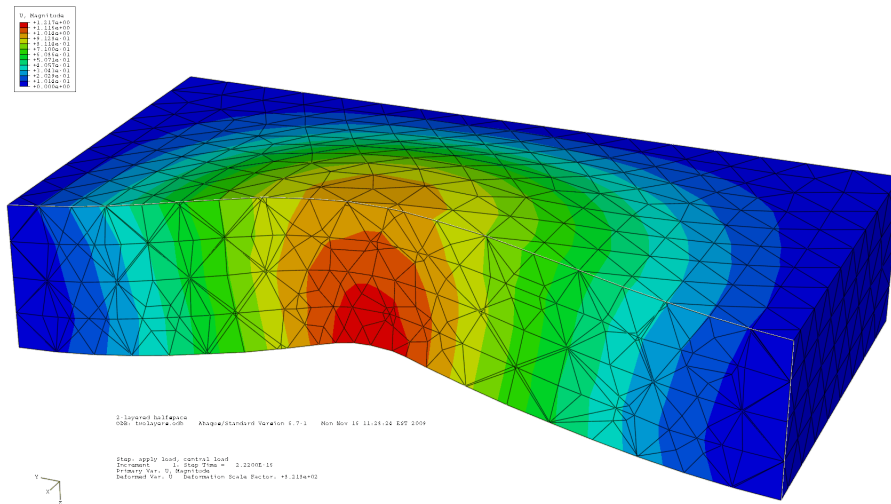


Figure 2: Example of how elements can be distorted in order to fit a geometry which will result in deterioration of gradient quality (image source: [15])

Localised metrics associated with the quality of each element have shown to be accurate for predicting the overall quality of a mesh when taking the average for each metric across all elements [14]. The quality of an elements shape is important since the stress values which are computed for the area within the element are calculated using the stress values at each of the nodes which enclose it [18]. Elements are typically deformed near parts of the geometry where its shape simply does not allow a uniform element to be placed, an example of where this has occurred can be seen in figure 2.

Some key shape metrics identified by Dittmer et al include (ideal values are for elements of quadrilateral type as used in the current prototypes):

- A Aspect ratio – longest side / shortest side, ideal value is 1
- B Maximum corner angle - widest internal angle to element, ideal is 90°
- C Maximum parallel deviation - how skewed the element is, ideal is 0°

4 Description of the work

4.1 Aims and Objectives

The aim of this project was to design, build and analyse a system for refining a mesh by combining a method from the domain of AI or machine learning with a stress based refinement method. The desirable end result will be a hybrid method of meshing which produces reasonably good solution accuracy whilst requiring less user intervention and reduced computational cost.

The project can be broken into three main sections of research and implementation which can each be considered a high level objective:

- A The first objective is to research and develop both a traditional stress refinement and an AI re meshing algorithm developed by industry or by academia. These algorithms should be able to run independently on a set of example models.
- B Secondly a process will need to be devised for combining the two re meshing methods to varying degrees will be required. This will make it possible to test the effects of a hybrid method across the same set of models. Through this it can be established whether there is notable benefit to combining the different approaches and if so to what degree is it effective.
- C The third objective will be to use justifiable metrics for assessing the quality of a given mesh. This will allow for a much greater range of hybrid variations to be tried without requiring inspection from an expert.

5 System specification

To demonstrate success in achieving the objectives of the project it is important to have tractability from the requirements through to the solution and lastly verification and validation. This section describes the current systems *functional requirements* (what the system will do) and *non-functional* requirements (its constraints) based upon evaluation of the research I have conducted in conjunction with discussions with the project supervisor: Dr Jason Atkin. Functional requirements have primarily been listed under their respective high level subsystems that are responsible for encapsulating their functionality.

Although it has not been developed as part of the project the application responsible for solving the finite element models has been included as part of the systems requirements since it highly influences the overall scope of the project and much of the design associated with other subsystems which were developed for the project.

6 Functional Requirements

1. FE integration: System will be able to interface with a third party finite element application
 - 1.1. The finite element applications solver must be able to solve a mesh based on its model configuration.

- 1.2. The finite element applications solver must be able to execute a model programmatically
- 1.3. The finite element applications solver must be able to output stress data at different points on the mesh.
- 1.4. The finite element application will provide a graphical representation of the model.
- 1.5. It will be possible to manipulate the model that the finite element application uses programmatically.
- 1.6. It should be possible to manipulate the model that the finite element application uses from within its graphical user interface.
2. Mesh refinement: System will be able to perform different kinds of finite element mesh refinement
 - 2.1. The system will be able to refine a finite element mesh using a stress based refinement method
 - 2.2. The system will be able to refine a finite element mesh using a non-stress based refinement method
 - 2.3. A non stress based refinement method will adapt the mesh using background information about mesh design which has been previously trained.
 - 2.4. The system will be able to combine the two methods to produce a coherent mesh which the FE application is able to successfully solve in order to obtain results for stress and displacement.
 - 2.5. The system will be able to combine both methods to varying degrees will be performed automatically by the system without direct user intervention.
 - 2.6. The system will re mesh using both stress and non-stress based refinement using quadrilateral elements.
 - 2.7. System will adapt weighting associated with each method based upon the metrics computer for the mesh in the systems previous iteration.
3. Quality assessment: System will provide the operator with results about the quality of meshes based on metrics obtained from research.
 - 3.1. An assessment will be conducted automatically for every mesh iteration that occurs.
 - 3.2. System will assess quality based on a variety of metrics to ensure overall robustness of measurement.
 - 3.3. The metrics will be computed for both individual elements within the model and for the entire mesh.

7 Non-Functional Requirements

Design: The system architecture will be developed using the object oriented design principals of SOLID to allow for clear interfaces between the different functional components. Functional programming practices will be adopted through use of the .NET Language Integrated Query or LINQ framework. This will help to simplify the code and improve reliability. Where functions and classes are written their length will be kept to a minimum to reduce complexity and allow for reuse wherever possible.

Documentation: The system will be comprehensively documented at both a code level and at an architecture level. At a code level C# doc comments will be written to provide a comprehensive summary of each function. This will allow the tool Doxygen [24] to generate a full set of developer documentation upon completion of the software implementation which will be included as an appendix. Doc comments will also help to encourage small functions.

General applicability: In order to demonstrate that hybrid methods are a feasible means of approaching meshing problems the resulting software should be able to successfully execute on a range of models with varying geometry. The range of geometries should be representative of typical structural variation.

8 System Design

8.1 System Overview

Determining the overall design of the system was initially hard since it was not clear exactly how many subsystems would be needed to mesh, evaluate and interface with LISA, what was clear was that the system would essentially be performing an optimisation procedure and as such needed to be driven iteratively towards a goal. It was therefore important to ensure the system maintained a modular architecture with well defined interfaces in order to allow for subsystems to be easily added and modified as the project progressed.

8.2 Modular Architecture

The modular architecture allows meshing algorithms and metrics for calculating mesh quality could be swapped interchanged was an important consideration in the design of the system. At best the quality of the output could be predicted for each method before it was integrated into the system and executed in a range of different scenarios. To have tightly coupled these individual components would have rendered the overall system a failure in the event that any one of them performed greatly worse than expected. Instead the loose coupling of the architecture has enabled the system to be considered as more of a framework for testing the effects of combining heuristic knowledge of a Finite element problem with a stress based refinement method.

8.3 Input Files

The system requires three basic input files which should be placed within a directory that is given to the program as a parameter, these are files are:

- A structural model represented as a .liml file which LISA can solve
- A
- A JSON file containing important edges as identified by an engineer along with associated meta data.

8.4 Multi Threading

To allow for better performance when evaluating a range of different weightings of the stress and heuristic methods multi threading was used to allow each configuration to be run independently of the others. When started each thread creates its own directory which it copies the three input files to and runs for its designating weighting configuration. Use of multi threading made assessment of the system much more rapid in the later stages of the project.

9 Software Implementation

The following subsections detail the implementation of the final software solution that has been written to meet the objectives posed at the beginning of this dissertation.

9.1 Third Party FE Application

In order to demonstrate the potential feasibility of the hybrid approach it was first important to obtain a finite element solver which could be given a FE model containing data about forces, materials and the mesh structure and then execute the model programmatically so as to obtain stress results.

A multitude of commercial FE tools exist with there being a wide variety in both the complexity and cost associated with each tool. Finite element software is typically very expensive due to its high development cost and small customer base. Tools used within industry such as ANSYS typically require a great deal of

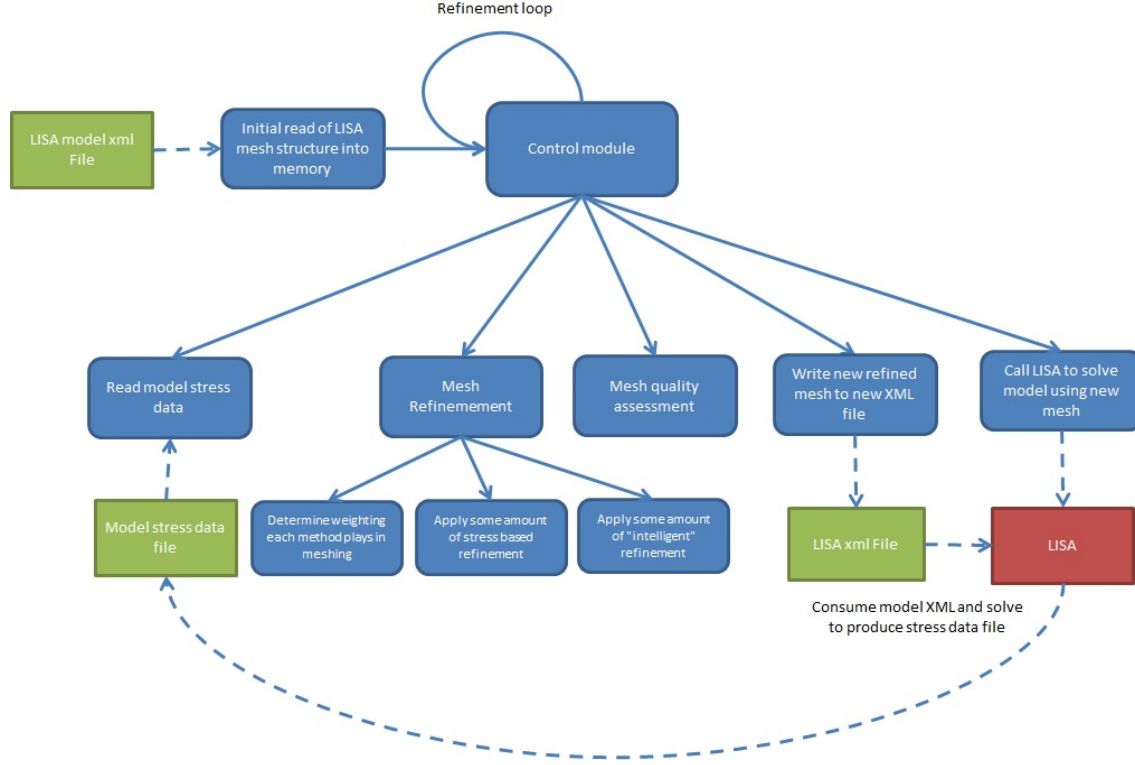


Figure 3: The basic bridge structure used to performing integration and system testing

time in order to become proficient in their usage and can cost in excess of five thousand pounds a year for a single licence [26]. It was therefore important to find a tool which was both affordable while also powerful enough to demonstrate a working prototype of the re meshing method.

9.2 LISA

After reviewing several FE applications used within industry in addition to a variety of less well known ones used within academia and by hobbyists LISA was selected as the solver application for which to implement the systems prototypes.

Strengths: LISA as a FE tool which allows the user to run models of up to 1300 element for free; This was beneficial in allowing me to experiment with the software and gauge the feasibility of my projects concept before requiring a full version. Once at a stage in the project where each problem had been solved for small models containing less than 1300 elements an academic licence for the software was purchased for the projects use.

LISA also provides a GUI which allows visual inspection of the model and its mesh; This is particularly useful for observing the output of the meshing algorithms which can often provide a human with a much better understanding of how the method has performed and whether or not there are obvious bugs. in the implementation of the meshing procedures

Weaknesses: Due to LISA's simplicity it does not come with an extensive API allowing for easy programmatic use of its inbuilt features, however it is still possible to interface with LISA through less direct means [19]. LISA models are stored in .liml files which use XML as a meta mark-up format. The model files contain all the information about the model including the materials used as well as loads and constraints

and of course the mesh. It is therefore possible to manipulate a .liml file having parsed its contents before writing a new version of the file which LISA can be called to solve. In order to more easily alter the model it made sense to write a wrapper for the .liml files to abstract the manipulation of their content.

9.3 Languages and platforms

The final system has been written entirely using the C# programming language (version 5.0) with Visual Studio 2015 as the development environment on a Windows 10 system. C# is an application development language built on the .NET framework. Although any number of programming languages could have been used to implement the solution C# offered a good compromise for developing a system with both structural rigidity through static typing and object orientation in addition to functionality to allow for rapid prototyping. C# does this well through use of LINQ, a part of the standard library that provides a large number of higher order functions which allow for operations to be performed over any data structure that implements the built in IEnumerable interface. Given that much of the code within the project performs the same operation on collections of nodes and elements stored in Lists arrays and dictionaries which all implement IEnumerable the ability to write much of the project using this capability dramatically reduced the number of errors encountered and increased development speed.

9.4 Implementation Methodology

Due to the size of the software, which is now in excess of 14000 lines of code it was important to work systematically in order to continuously drive the project in the right direction and avoid the introduction of unnecessary complexity which would likely have resulted in the creation of bugs or simply increased the difficulty of adding new features.

For the duration of the project the spiral methodology was adhered to. This enforced multiple deliverable stages for the project which were concluded with a supervisor meeting every one or two weeks. Adopting the spiral methodology also provided flexibility regarding the order in which tasks were able to take place outside of a spiral iteration. This was useful is necessary when conducting a research project where direction of work for subsequent iterations was largely driven by the findings of the work in the current iteration.

Tasks were chosen every week for the project, the number of tasks and their complexity was determined using a combination of factors including their complexity, the criticality of the task e.g. did it need to be completed for other important tasks to be started and the time available to me as the individual undertaking the project (More tasks typically performed on weeks when less work was due for other modules.)

9.5 Implementation of Subsystems

This section describes the implementation details of the various different subsystems which combine to form the overall solution.

9.5.1 Re-meshing using hierarchical refinement

Elements within FEA can typically be classified as either triangle or square based elements, each of these provide different strengths and weaknesses when required to mesh and solve models. Within industry triangular elements are typically preferable since it is always possible to generate an initial triangular mesh from any arbitrary CAD geometry algorithmically. This is done by simply making smaller triangles until all gaps along the edge of the geometry are filled []. The same cannot always be said when meshing using square elements. For proof of the solutions concept however it was concluded that square based elements were preferable to triangular ones with since the steps required for a basic refinement are much simpler. In addition to this Its also significantly easier to define edges which the ILP rules can be applied to when edges naturally form within a structure through a chain of nodes along the edges of square elements.

After reviewing both h-refinement [2] and r-refinement [3] it was concluded h-refinement would be best the best approach to adopt for use in a due to its simplicity and more widespread use [2] despite the fact

that the mesh is usually more computationally expensive than it would have been if it was created using r-refinement [3]

Unfortunately Triangular meshes also generally incur a higher computational cost than an equivalent square element mesh due to added complexity of performing the calculations required to remesh in addition to requiring more elements over a given area to achieve the same accuracy.

From an implementation standpoint writing a square based remeshing algorithm is also substantially easier as given an initial mesh made just of square elements the only task is to repeatedly divide each element into four sub elements, by contrast methods used to remesh triangular meshes vary greatly and are typically more complex and have corresponding initial meshes that are harder to generate manually by human operators [?].

9.5.2 Stress Based Refinement

To focus meshing in areas of high stress each iteration needed to parse the results file from the previous iterations execution of LISA. LISA result files are in csv format by default and contain the displacements and stresses associated with each node within the model once it has been solved.

Once the data in the output file has been parsed meshing can be conducted for high stress areas by averaging the stresses across the whole model and then cross referencing the node Ids in the results which have stresses above the average against the main data model allows a list of elements that can be instructed to refine themselves.

9.5.3 Rule Based Refinement

Having successfully implemented an simple stress refinement method the next key step of the systems development was to write a framework for executing each of the ILP rules published by Dolsak in his papers [4, 5, 6] [7].

Each rule is represented by a function within the final solution, this closely resembles their representation by Dolsak in his papers. Each of the rules resides within the "RuleManager" class and takes a number of the defined edges as parameters. Each rule then checks the properties of a particular edge against properties which have been identified through the ILP learning mechanism as being important when the model executes. In cases where the rule takes more than one edge as an argument comparison for the properties of each edge are performed. Where a rule detects a relationship in the model the edge is assigned a value corresponding to its criticality, the value is then used by the meshing procedure to determine how many times it should repeatedly remesh the elements along that edge.

the significance of elements along a particular type of edge in determining the accuracy of the model. Some rules look for highly critical relationships which if found have a high importance assigned to them,

indicate that additional meshing needs to occur for all elements that border the edge. The amount of meshing that occurs for each of the elements along those edges is specified within each of the rules.

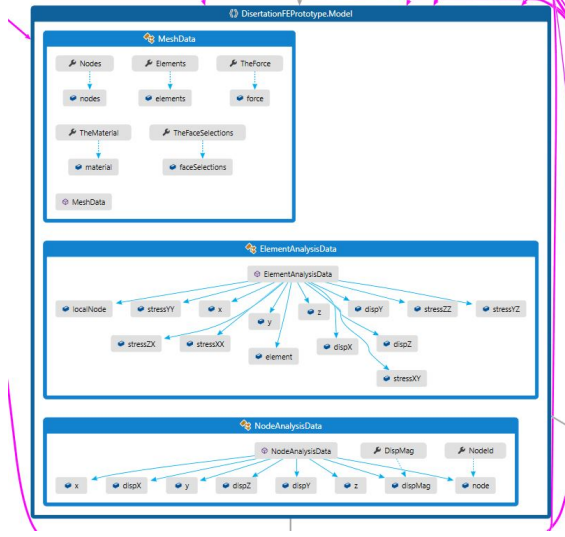
The properties that can exist between two edges when compared are the following:

- Edges opposite one another - the edges run alongside one another closely
- Edges possess the same form -
- Edges are considered the same - to meet this requirement both edges must be almost the same length, opposite one another and possess the same form.

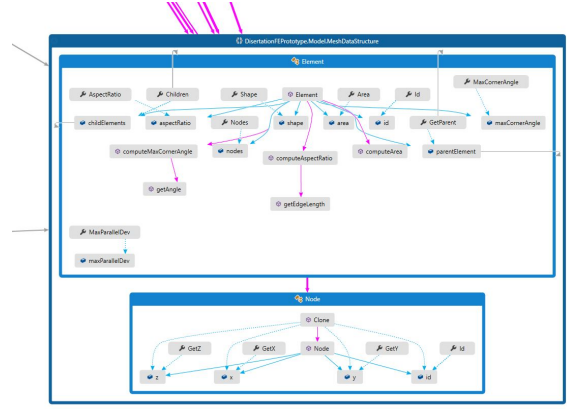
9.5.4 Interfacing with LISA and the FE data model

Writing an API for LISA was the first stage of development for my project. The API was crucial in order to which rely on the existence of different operations in order to easily manipulate the mesh. When the first re-meshing iteration occurs the system reads the .liml file into an equivalent class model, diagrams for

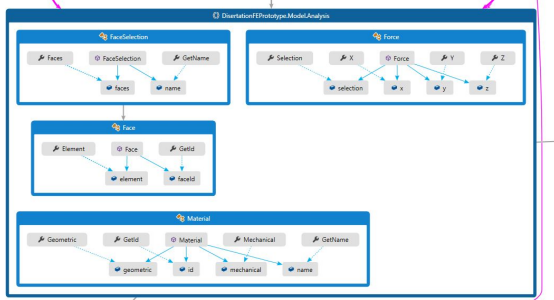
which can be seen in figure 4 below. Each class in this model contains corresponding data and methods used to represent and manipulate the model. These methods are then used by the different refinement methods to generate a mesh which is ideally more optimal for the given problem, this adapted mesh can then be assessed by the module responsible for validating the meshes quality and written back to a .liml file for LISA to solve on the subsequent iteration. When designing the data model to represent the FE models state it was desirable to maintain coherence with the structure specified by LISA within the input XML files as closely as possible, this allowed for the data model to more easily be serialised when communicating with LISA.



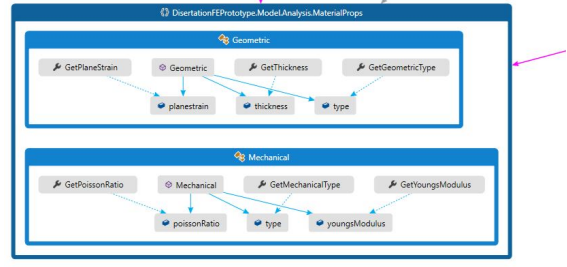
(a) Model classes



(b) Element and Node classes



(a) Model Analysis classes



(b) Material Property classes

Figure 5: Class model to represent .liml file structure used by LISA

One aspect of the data models design which greatly adds to the systems flexibility is the hierarchical design for representing the various Element types. At the root of this structure is the IElement interface, all new Element types must adhere to this in order for the various refinement methods to run. Implementing the interface are a range of abstract classes such as "SquareBasedElem" and "TriangleBasedElem" These classes are designed to contain methods that are generally applicable for calculating metrics and re meshing individual elements for elements of this type that are both in two and three dimensions. This is powerful since computing this for a 3d element is simply a reduction of the method for a 2d element over every face which comprises the 3d one. Finally underneath this level are the concrete classes which represent each specific type of element that can exist.

9.5.5 Mesh Quality Assessment

Dittmers rules for computing the quality of both individual elements and the entire mesh are built into their own "MeshQualityAssessments" and "ElementQualityMetrics" classes, the latter of which is encapsulated within an element object, like with refinement this allows each element to assess its own quality removing the need for additional utility classes containing static methods.

Since each element is initialised with the nodes that comprise it, it is also possible to derive all the geometric characteristics and thus its quality metrics upon its initialisation. This allows the metrics for each node to also be calculated upon its initialisation removing the risk of elements returning null when asked for them.

9.6 Modelling Real World problems

Although the main implementation for this project was the system capable of performing the tasks described above on very simple structures such as a cantilever beam, to demonstrate general applicability with regards to real engineering problems demanded the creation of several models resembling basic equivalents of structures that FE methods would be used on within real engineering analysis.

9.6.1 Suspension Bridge structure

The primary model for which work on the latter half of the project was based upon was the following suspension bridge model:

The initial model which was constructed manually using LISA's graphical user interface given some basic measurements for such a bridge in real life consists of 196 elements and 212 nodes, which is very coarse for a FE mesh structure. The bridge is constrained at the base of each pillar, meaning all forces must exit the model through one of the four pillars. When

One of the bridge models strengths was it provided high versatility when testing the software solution for m since simulation could be run with forces placed on a range of different faces with corresponding edge rules and a range of varying results obtained. For example a force on the deck of the bridge could be used to simulate the weight of a vehicle moving over it while forces placed sideways on the towers and cables could be used to model stress induced from strong crosswinds,

both of these scenarios when used to evaluate the effectiveness of the implementation

9.6.2 Validating system against previously obtained research results

Due to a lack of specific implementation details provided by Dolsak and Muggleton in their papers for generating meshes it was important to demonstrate that the underlying implementation was essentially equivalent.

To do this Models produced by Dolsak used to train his initial ILP system which generated the rules running within my system were created. This allowed my to run my resulting system on the models and verify that the generated output resembled those published in the current research literature [4].

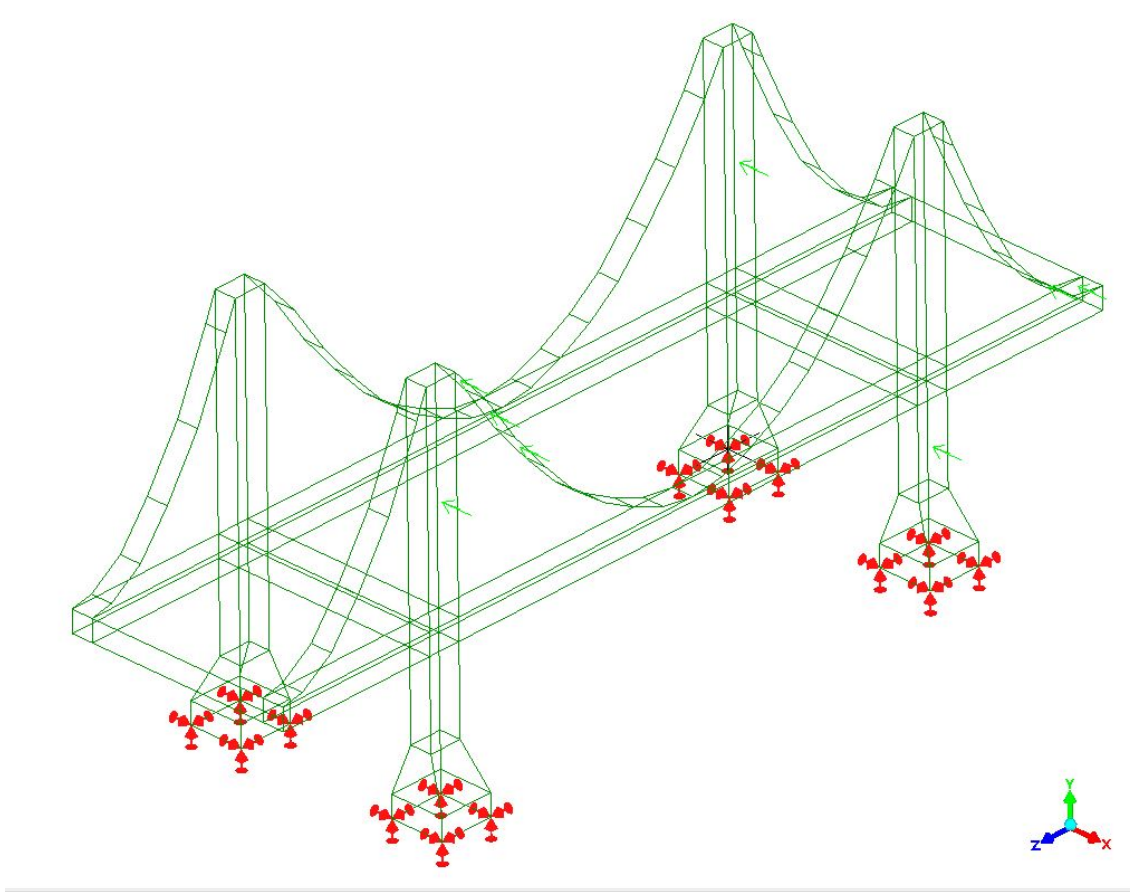


Figure 6: Diagram representing the general structure of the system with interactions between non internal entities such as files and LIA shown using dashed lines

9.7 Implementation Issues

Implementation of the design was not without its difficulties, many of which arising as a consequence of unforeseeable complications when implementing the well understood theoretical aspects.

9.7.1 Sorting Element Nodes in 3d space using convex hull Algorithm

A significant issue encountered when working with LISA on the project was an interface requirement specified by LISA requiring nodes for each type of element to be sorted in a specific geometric order. The general rule for node ordering within LISA is to order the nodes as a perimeter around an element in 3d space without paths between nodes crossing one another internal to the element. When first addressing this problem for simple models using elements of Quad4 type the obvious approach was to think of a simple quadrilateral resembling a square, write an algorithm to handle that as a base case before considering a more complex case. The resulting approach was the following:

This approach was sufficient for the vast majority of elements within the different models, however as model complexity increased multiple iterations of the heuristics resulted in increasingly distorted element shapes, resulting in rejection of elements with incorrectly sorted nodes by LISA.

In order to resolve this issue I needed to research and apply a convex hull algorithm to this problem. The subsequent solution was the following:

analyzed. The first example was the cylinder in Fig. 2 [4], the second example was the hook in Fig. 4 and the last example was the cross-section of a paper mill in Fig. 5 [5].

Golem needs three types of input files to build the rules:

- foreground examples,
- negative examples,
- background facts.

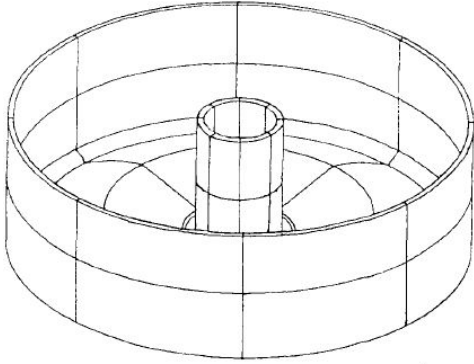
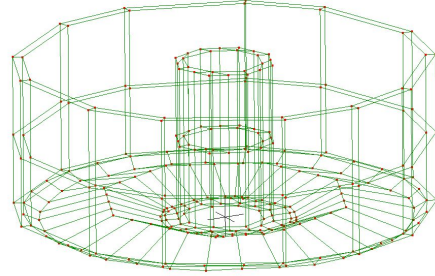


Fig. 5. Training example: the paper mill.



[!h]

(b) Representation of basic paper mill within LISA before applying meshing rules

[!h]

(a) Paper Mill input provided by Dolsak

Figure 7: Class model to represent .liml file structure used by LISA

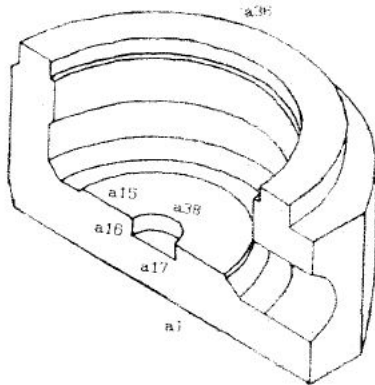


Fig. 3. Labelled edges of the structure in Fig. 1.

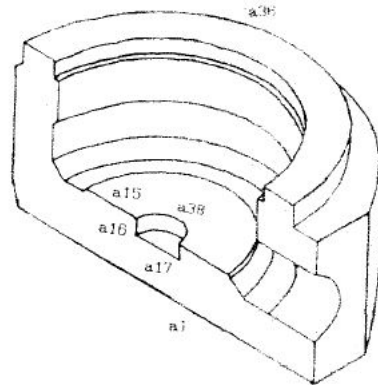


Fig. 3. Labelled edges of the structure in Fig. 1.

[!h]

(a) Cylinder model used by Dolsak with edges labeled, each labelling corresponds to a rule generated by the Golem algorithm [4]

[!h]

(b) Diagram representing the general structure of the system with interactions between non internal entities such as files and LIA shown using dashed lines

Figure 8: Class model to represent .liml file structure used by LISA

Figure 9

```

unsortedNodes = Node[1...4];
sortedNodes = Node[1...4];
while number of sorted nodes < number of nodes in element do
    Add first node in list that needs sorting to sorted list;
    foreach s ∈ unsortedNodes do
        compute distance from most recently sorted to s;
        add to
    end
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end

```

Algorithm 1: Initially Improvised approach

Despite the existence of algorithms for generating a convex hull in three dimensions due to the small number of points it made sense to assess the effectiveness of a 2d method first before committing and assess the effectiveness of that for resolving my problem. In order to do this I simply took my Quad4 elements and flattened them to a 2d representation by calculating the maximum delta between the greatest and smallest value on each axis and eliminating the axis with the smallest delta. This proved successful, when applied to the model this successfully removed incorrect ordering from elements that had been particularly skewed.

Data: this text
 Flatten element to two dimensions;
while do
 read current;
 if *understand* **then**
 go to next section;
 current section becomes this one;
 else
 go back to the beginning of current section;
 end
end

Algorithm 2: Convex Hull Method

This method has $O(n \log n)$ time complexity however due to the size of n being 4 in all cases the complexity of sorting an individual element is constant, with the overall complexity of sorting all elements in the model being $O(n)$ where n is the number of elements.

9.7.2 Attempts to Automatically Define edges within models

As discussed under evaluation one of the core issues faced when performing analysis of the system is identifying where the system behaves poorly through weakness in the methods or poor design and implementation as opposed to poor output generated as a consequence of poor input by the operator. In order to avoid this issue it was desirable to try and remove user intervention besides the configuration of the initial model. The obvious way by which to do this was automatic identification of interesting edges within the model. A crucial property which made this approach appear promising is the fact that it is known that edge importance

directly correlates with the size of the edge and how much force is applied near it, since both this information exists within the data model it should be possible to identify edges from it and generate them automatically for Dolsaks rules to process.

In practice there are multiple complications to doing this however, several of these arise from ambiguity in Dolsaks paper regarding what constitutes an edge which is for example "long", "important" and does not provide clear definitions for

Having considered a method using these properties of the mesh several days were then spent attempting to implement it.

Having done so

9.7.3

10 Evaluation of Project

Evaluation has been essential in order to conclude the effectiveness of both the algorithms used to solve the problem and the capability of their implementation within the final software solution. In order to do this evaluation has been conducted at several levels ranging from functional requirements for the software specification at the lowest to high level analysis of the programs output which in many cases is hard to define and varies from model to model.

11 Functional testing

In order to validate the system against many of the functional requirements the system only needs to be run on several basic models with different input configurations, observation of the output demonstrates the systems ability to evaluate the quality of meshes using a range of metrics and refinement occurring as a result of both the stresses induced by the user and on their categorisation of edges.

12 Non functional testing

12.1 Increase in performance through parallel execution

The following

The average speed up ($\text{Time of serial execution} / \text{Time of parallel execution}$) was

The Efficiency of the parallel execution ($\text{speedup} / \text{number of processors}$) with an Intel i5 processor with 4 cores the average efficiency was calculated as

12.2 Quality

The quality of the design and implementation of the system reflects my ability not only as a computer science undergraduate but as a developer with one year industrial experience, although not directly effecting the execution of the program properties such as appropriate variable naming, loose coupling of classes, use of abstractions and descriptive error messages make the software easier to read and debug for any potential future developers.

12.3 Documentation

The process of continuously writing descriptive documentation was important to the success of the project and was treated as an integral part to meeting to the goals of the project development methodology which aimed to reduce the systems complexity and improve readability. Through the writing Doc comments corresponding to every function within the codebase it was possible to generate documentation files automatically

through use of the tool Doxygen. This allows anyone with the solution to view descriptions of each of its functions either in the codebase or alternatively through the manual produced automatically by Doxygen.

12.4 Maintainability

As a result of loosely coupled design with each file containing only one class and each class and function written with the goal of providing only one item of functionality, the
and well written documentation the system is for the most part maintainable.

12.4.1 Evaluation Issues

A significant issue faced in attempting to demonstrate the effectiveness of the system was to provide an indication of how well the system worked without taking into account the ability of the user who may be providing the edge rules for a particular model. Not taking this into account would result in an inaccurate representation of its ability.

12.5 Evaluation of overall project

12.6 Evaluation summary

12.7 Successes and Limitations

13 Further Work

This section details some areas which given additional time to work on the project would at the very least be investigated, if not implemented. Each of these areas would hopefully provide some benefit in assisting to demonstrate the possibilities of hybrid methods.

13.1 Gathering feedback from experienced engineers

Approaching the end of the project it became clear that in order to better identify the systems strengths and weaknesses would require additional user testing by engineers who have experience conducting this type of analysis. Despite not having the available time to conduct user feedback I maintained contact throughout the project with a variety of acquaintances I was able to make through my industrial placement year at gas turbine manufacturer Rolls-Royce, with several expressing interest in the concept. Combining feedback from engineers with extensive applied industrial experience and that of academics within the universities own mechanical engineering department would hopefully have allowed for a more conclusive analysis of the systems overall capabilities. Obtaining user feedback was not considered feasible given the amount of time to implement the project and the already inherent complexities in producing a working implementation for just a few test cases. As such even if time had been available the ethical clearance required to collect user feedback at the start was not obtained.

13.2 Improving usability through a web interface

Although possible to visit various engineers in order to conduct feedback the process is both time consuming on my part and inconvenient for the participant as a rigid time for which to meet must be scheduled and a laptop containing the working software brought to them which they must design or transfer their model to before running it multiple times to obtain results. This scenario is at best inconvenient for the participants and pressures them into arriving at a conclusion within a relatively small time of experimenting with it.

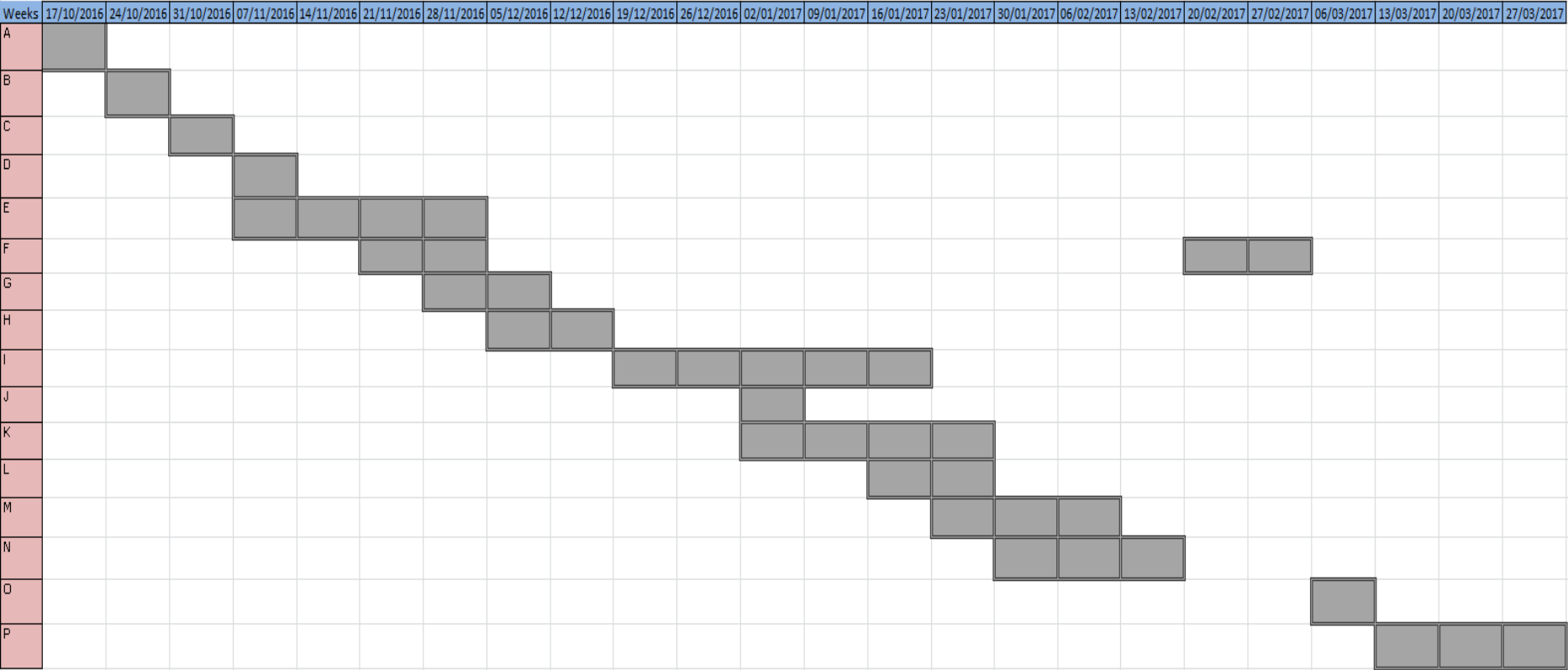
Instead by facilitating interaction with the system by means of a web interface the engineers would be able spend as little or as much time as they like experimenting with the system and allow them to submit feedback digitally. This would potentially allow feedback to be obtained from a much wider range of different sources

separated by significant geographic distance.

In order to theoretically use such an interface each engineer would simply have to download the free version of LISA from the following webpage <http://www.lisafea.com/index.html> import a model they already have which is less than 1300 elements(LISA supports imports from multiple CAD formats including Standard for the Exchange of Product model data (STEP) and Initial Graphics Exchange Specification (IGES)) [?] and then submit this along with a JSON file containing edges they have designated as important for their model. Upon receiving the request the web server would the current project with their input data and having finished allow them to download the re meshed model along with the calculated stress data for analysis.

14 Personal Reflections and Summary

Comparing the current progress of the project against the plan things are on track to be completed as scheduled. So far I am pleased with both my research efforts and the functionality that I have been able to implement. My primary concern at present is that the ILP generated rules may not perform as well as expected when fully implemented or that they perform well but only for a limited set of geometries, which would be disappointing. While conducting research, development and implementation for the project I have worked methodically to best understand each problem as and when they occur and consider each of the possible solutions before committing to one. I feel this approached has saved me much time and has forced me to reach a better view about what exactly each subsystem should do and how to implement it. In hindsight if I were to repeat the first part of the project again I would have organised my time differently to spend less of it focussed on the details of coursework assignments for other modules into making further progress on the implementation of the rule system.



References

- [1] Max D. Gunzburger, Janet S. Peterson *Finite Element Methods* https://people.sc.fsu.edu/~jburkardt/classes/fem_2011/chapter1.pdf
- [2] Adaptive Finite Element Techniques <http://www.cs.rpi.edu/~flaherje/pdf/fea8.pdf>
- [3] Scott McRae *r-Refinement grid adaptation algorithms and issues*
- [4] Bojan Dolsak and Anton Jezernik *Mesh generation expert system for engineering analysis with FEM*
- [5] Bojan Dolsak, Anton Jezernik *A knowledge base for finite element mesh design* Artificial Intelligence in Engineering 9 (1994)
- [6] Bojan Dolsak, Stephen Muggleton *The Application of Inductive Logic Programming to Finite Element Mesh Design*
- [7] Bojan Dolsak, Frank Reig, Reinhard Hackenschmidt *Consultative Rule-Based Intelligent System for Finite Element Type Selection* Research Gate 2016
- [8] Paul Dvorak *Two meshing methods are better than one* <http://machinedesign.com/archive/two-meshing-methods-are-better-one>
- [9] Larry Manevitz, Malik Yousef, Dan Givoli *Automatic Mesh Generation (for Finite Element Method) Using Self-Organising Neural Networks*
- [10] Abid Ali Khan, Imran Ali Chaudhry² & Ali SaroshCase *Case Based Reasoning Support for Adaptive Finite Element Analysis: Mesh Selection for an Integrated System*
- [11] Stephen Muggleton *Inductive Logic Programming*
- [12] <http://www-ai.ijs.si/~ilpnet2/systems/golem.html>
- [13] Stephen Muggleton *Logic based and Probabilistic Symbolic Learning* <https://www.youtube.com/watch?v=4Cwd05dWW98>
- [14] Jeremy P. Dittmer, C. Greg Jensen, Michael Gottschalk, and Thomas Almy *Mesh Optimisation Using a Genetic Algorithm to Control Mesh Creation Parameters*
- [15] <http://danielpeter.github.io/rays.html>
- [16] Lina Vasiliauskiene, Romualdas BAUÅÏYS *Intelligent Initial Finite Element Mesh Generation for Solutions of 2D Problems* INFORMATICA, 2002, Vol. 13, No. 2, 239aÅ\$250 2002
- [17] E.Bellengera,Y.Benhafidb, N.Troussierb *Framework for controlled cost and quality of assumptions in finite element analysis* Finite Elements in Analysis and Design 45 (2009) 25–36
- [18] G. P. Nikishkov *INTRODUCTION TO THE FINITE ELEMENT METHOD* <http://homepages.cae.wisc.edu/~suresh/ME964Website/M964Notes/Notes/introfem.pdf>
- [19] <http://www.lisafea.com/pdf/manual.pdf>
- [20] Nam-Ho Kim *STRUCTURAL DESIGN USING FINITE ELEMENTS* http://web.mae.ufl.edu/nkim/eas6939/Opt_FEM.pdf
- [21] *Type of Finite Elements and Steps in FEA Process* <http://highered.mheducation.com/sites/dl/free/0073398144/934758/Ch07TypesOfFiniteElementsAndStepsInFEAProcess.pdf>
- [22] <https://www.quora.com/What-is-the-cantilever-beam-What-is-the-advantages-and-disadvantages-of-it>
- [23] <http://www.lisafea.com/purchase.html>
- [24] <http://www.stack.nl/~dimitri/doxygen/>
- [25] <https://caeai.com/blog/will-poorly-shaped-elements-really-affect-my-solution>
- [26] <http://mscnastrannovice.blogspot.co.uk/2013/04/how-much-does-ansys-cost.html>
- [27] <http://www.engineeringanalysiservices.com/moving-mesh-fea-analysis.php>
- [28] <http://loyc.net/2014/2d-convex-hull-in-cs.html>