School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

Final Report

Learning Resources for Understanding and Using the Finite Element Method

Kyle Antony Morris

Submitted in accordance with the requirements for the degree of Computer Science with Mathematics

2022/23

COMP3931 Individual Project

The candidate confirms that the following have been submitted.

Items	Format	Recipient(s) and Date
Final Report	PDF file	Uploaded to Minerva (DD/MM/YY)
<Example> Scanned participant consent forms	PDF file / file archive	Uploaded to Minerva (DD/MM/YY)
<Example> Link to online code repository	URL	Sent to supervisor and assessor (DD/MM/YY)
<Example> User manuals	PDF file	Sent to client and supervisor (DD/MM/YY)

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

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of Student) ____________________

Summary

<Concise statement of the problem you intended to solve and main achievements (no more than one A4 page)>

Acknowledgements

<The page should contain any acknowledgements to those who have assisted with your work. Where you have worked as part of a team, you should, where appropriate, reference to any contribution made by other to the project.>

Note that it is not acceptable to solicit assistance on ‘proof reading’ which is defined as the “the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the test”; see
`https://www.leeds.ac.uk/secretariat/documents/proof_reading_policy.pdf`

Contents

Chapter 1

Introduction and Background Research

1.1 Introduction

Partial differential equations (PDEs) are some of the most important types of formulae in mathematics. They arise in every field of mathematically inclined science, such as physics, engineering, chemistry and more. PDEs are used to describe physical systems. Some PDEs describe dynamical systems; that is, systems that change with time. Others describe static systems, that only vary in space, or sometimes other quantities. This might include the movement of fluids, heat and waves, or the deformation of certain structures acting under a force. They are used to model population dynamics, chemical reactions, electromagnetic fields and are also central to quantum mechanics. PDEs are fundamental to our understanding of science and the world[1].

Since PDEs are so important, the natural question is, how are they solved? Solving a PDE simply means obtaining an equation that describes the key variable explicitly, e.g. velocity of fluid particles given their coordinates in space, number of individuals in a population at any time, or amount of heat at a certain point on a metal rod. There are methods for solving certain types of PDE analytically, but generally PDEs are either incredibly difficult to solve, or in many cases, impossible.

Fortunately, methods have been developed to provide approximate solutions to PDEs, called numerical methods. These methods always have some degree of error, but by increasing the amount of computation performed, this error can be reduced. Numerical methods can involve an enormous amount of computation, making them the ideal candidates for modern computing, where processors perform billions of operations per second.

One such method is called the Finite Element Method (FEM), where the domain being solved over is discretised, and linear algebra techniques are used to construct a linear system of equations that can be numerically solved to obtain our approximate solution[2]. The method has a strong mathematical foundation and it can be shown that the approximation does converge as the number of discrete elements increases[3]. The method is fantastically versatile, as it can be applied to any PDE, whether linear or non-linear, the only problem is that it requires a good mathematical understanding to implement.

The aim of this project is to address this issue and create a set of learning resources that can be used to bridge the gap in understanding required to use FEM. The goal is to guide students through the mathematics as well as instruct them on how to implement the method themselves. In this report I will describe the background research done to influence my design decisions, detail the writing process, and discuss the results obtained from user feedback.

1.2 The Finite Element Method

1.2.1 How the Finite Element Method Works

As previously mentioned, FEM is a numerical method for solving differential equations. Most often, it is applied to partial differential equations. The general form of a PDE is shown in Equation 1.1. Here, the variables are denoted $t, x_1, \cdots, x_n$, of which there are $n+1$. The order of the equation is the order of the highest derivative of u, denoted m. u is the unknown function that being solved for. Note that t may not always be present here.

$$
\begin{aligned}
F \Big(& u, t, x_1, x_2, \cdots, x_n \\
& \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \cdots, \frac{\partial u}{\partial x_n}, \\
& \frac{\partial^2 u}{\partial^2 x_1^2}, \frac{\partial^2 u}{\partial x_2^2}, \cdots, \frac{\partial^2 u}{\partial x_n^2}, \\
& \qquad \cdots \\
& \frac{\partial^m u}{\partial x_1^m}, \frac{\partial^m u}{\partial x_2^m}, \cdots, \frac{\partial^m u}{\partial x_n^m} \Big) = 0
\end{aligned}
\tag{1.1}
$$

Clearly, Equation 1.1 is a complicated equation, but most PDEs do not have this many variables. Also, it is rare that a PDE will have an order greater than two, although this can occur. Along side the equation itself, there will also be a set of conditions that constrain the variables to some domain Ω. There will also be a set of conditions that constrain u on the boundary of this domain $\delta\Omega$. These are called the boundary conditions. If one of the variables involved is time this must also be constrained by an initial condition. Together these make a initial value boundary problem (IVBP). A full instance can be seen in Equation 1.2.

$$
\begin{cases}
F = 0 & \text{on } \Omega \\
u = u_D(x_1, \cdots, x_n, t) & \text{on } \delta\Omega \\
u = u_{\text{init}}(x_1, \cdots, x_n) & \text{at } t = 0
\end{cases}
\tag{1.2}
$$

$$\text{where } \Omega = [0,1] \times [0,1]$$

The method of solving such an equation using the finite element method first starts by converting Equation 1.2 into a variational formulation. Specific methods of constructing this formulation vary from problem to problem but the general approach is as follows:

- Take the equation defined over Ω and multiply both sides by a test function v.
- Integrate both sides over Ω.
- Integrate any second order derivatives by parts.
- Rearrange so all terms involving u are on one side. This expression is called the bilinear form, denoted $a(u, v)$ and the other side is called the linear form, denoted $L(v)$.

After doing this, a few facts about the functions spaces u and v belong to need to be acknowledged. The class of functions that can solve Equation 1.2 belong to the function space,

$$V = \{v : v \in H^1(\Omega), v = u_D \text{ on } \delta\Omega, v = u_{\text{init}} \text{ at } t = 0\}$$

Clearly, in addition to this, they must also satisfy the main equation. $H_1(\Omega)$ is the Sobloev space, which is a space where all functions must be square integrable and continuous and their derivatives must simply be square integrable. V is called the trial space, and u is called the trial function. In order for the variational formulation to be valid, the test function must also come from $H_1(\Omega)$, but instead include the condition that it must vanish on the boundary of our domain. This yields the test space,

$$V_0 = \{v : v \in H^1(\Omega), v = 0 \text{ on } \delta\Omega, \}$$

This trick allows us to remove any parts of our integration that include v on $\delta\Omega$. More importantly however, subspaces of V and V_0, denoted V_h and $V_{h,0}$ can be constructed, where the elements are piecewise polynomial functions. This can only be done because these spaces permit functions with discontinuous derivatives. So the transition from a continuous space to a discontinuous one is made, replacing u with $u_h \in V_0$ and assuming $v \in V_{h,0}$. This is the finite element approximated problem.

$$\begin{aligned} &\text{Solve for } u_h \in V_h \text{ where,} \\ &a(u_h, v) = L(v) \quad \forall v \in V_{h,0} \end{aligned} \tag{1.3}$$

To solve this problem, our domain Ω must be split into discrete segments, called a mesh. In one dimension, this is amounts to splitting the number line into sub-intervals, in two dimensions, the plane is split into cells, or finite elements. These cells are primitive shapes like rectangles or triangles and depending on the domain, either it can be simpler to use one or the other. The points where these cells meet each other are called nodes.

Now to solve, techniques from linear algebra are used. A basis of V_h and $V_{h,0}$ must be constructed to rewrite u and v as a linear combination of basis functions. The one used in the finite element method is called the nodal basis, because the coefficients of a linear combination of each of these basis vectors are equal to the function at the nodes of the mesh. The usefulness of this, is that any function can be defined using only these nodal values, meaning if they can be obtained, the approximate solution can be found.

The way to obtain these coefficients, called degrees of freedom, is by substituting in u and v as a linear combination of these basis functions, called hat functions. This produces $n + 1$ independent equations, where n is the number of nodes in the mesh. Then, the resulting system of equations can be solved using known numerical methods. One can use direct methods like Gaussian elimination and LU-Factorisation, or where appropriate, iterative methods like Jacobi, or Gauss-Seidel.

If time is a variable in the equation, one can use the numerical methods used for ordinary differential equations to discretise the time domain and obtain an approximation that way. The Runge-Kutta methods are a popular choice.

1.2.2 Convergence and Error Estimates

FEM is an extremely powerful technique because it can be applied to any problem of the form 1.1 if a variational form can be defined, and it can be shown that all problems have such a formulation [4]. There may be some practical difficulties along the way, but the sophistication of simultaneous equation solvers means that the most computationally intense problems can be scaled to work on high-performance clusters.
One natural question that arises is that of convergence. Convergence of the method can be proved by showing that the error of our approximated solution tends to 0 as the mesh size increases. A natural way to measure error is using the L^2-norm. This is because the Sobloev space H^1 is actually a Hilbert space, defined with the L^2-norm as its inner product. Thus, the L^2-norm is defined for all elements of V_h. The L^2 inner product of two functions f and g is defined as

$$\langle f, g \rangle_{L^2(\Omega)} = \int_\Omega f\bar{g}\,\mathrm{d}x$$

Where $\bar{g}$ is the complex conjugate of g. Then, the L^2-norm is the square root of the inner product of a function with itself, denoted

$$\|v\|_{L^2(\Omega)} = \sqrt{\int_\Omega |v^2|\,\mathrm{d}x}$$

An enlightening example would be $\mathbb{R}^n$, which is also a Hilbert space when equipped with the familiar "dot" product as its inner product. The norm can then be though of as the distance between two points if the following is calculated,

$$\|x - y\| = \sqrt{(x-y)\cdot(x-y)} = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

This is precisely the definition of distance between two points. H^1 works the same way, only with a different notion of "distance", the L^2-norm $\|u - v\|_{L^2(\Omega)}$. Then, the error of our approximation u_h from the actual solution u can be measured by $\|u - u_h\|_{L^2(\Omega)}$. In Bengzon and Larson's textbook on the finite element method[3], they find the best estimate of error to be

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^2\|D^2u\|_{L^2(\Omega)} \tag{1.4}$$

Where C is a constant, D^2u is the second order total derivative of u and h is defined to be the longest edge of any cell in the mesh, which is just a way of measuring the mesh size. It is clear from 1.4 that as $h \to 0$, the error also tends to 0, which means the finite element converges as the size of the cells in the mesh decreases.

1.2.3 Overview of Tools

The process outlined in Section 1.2.1 is difficult to implement without any libraries, even for the simplest of problems. This is due to the mathematical complexity of all the operations that would need to be performed. The functions one would need to implement include numerical integration, evaluating hat functions, partial derivatives, linear algebra solvers, and much more. This is why, most people prefer to use tools in order to perform finite element analysis. These tools come in the form of libraries and modules, to bespoke programs. Each have their benefits and advantages, and we will analyse these now.

1.3 Overview of Literature Regarding Mathematical Education

** Discuss different taxonomies, best practices, and approaches. Reference empirical studies if possible. **

1.3.1 Educational Practices

In order to write the learning resources effectively, a concrete understanding of the Finite Element Method is needed. However an equally, and often forgotten part, of writing a good learning resource involves having a good knowledge of education and the utilisation of an approach that puts the student experience first.
"How to write a good textbook / learning resource - blooms taxonomy, learning objects"

1.3.2 Types of Learning resource

"Textbooks, in-person courses, video, blended learning, e-learning and mobile learning in mathematics education"

1.3.3 Application to Mathematics

"Khan academy, brilliant, codecademy"

1.4 Overview of Existing FEM Learning Resources

** Review a few online tutorials, discuss what is good and bad about each with respect to the previous subsection. Do the same for books. Also a good point to review some of the materials I want to base the notebooks off of and explain why I chose them. **

1.4.1 Book by Brenner-Scott

1.4.2 Book by Bengzon-Larsen

1.4.3 Tutorial by Jorgen S Dokken

Chapter 2

Methods

2.1 Justification of software choices

** Discuss Jupyter Notebook as a learning tool and why it makes for a good solution with respect to the background research performed. Talk about: why Python is good for numerical computation and good for people with a mathematical background; why DOLFINx is a great package for implementing the Finite Element Method and what dependencies come with that; how version control is used and the benefits of good version control management **
Maybe use Binder to run the repo? https://mybinder.org/ Or possibly nbviewer, although this does not allow you to run Github Pages
Discuss the different technologies above and why I went with Binder. Discuss accessibility, lowering the barrier to learning.
https://computational-acoustics.gitlab.io/website/posts/30-intro-to-fenics-part-1/ Pros and cons of FENICSx

2.2 Justification of design choices

** Talk about content plan, why I structured it the way I did and why I chose the content I did, always referring back to relevant research. Motivate the need for a mathematical understanding of the problem and it's solution instead of simply solving using packages you don't understand. This section should talk more about the education and mathematics, and should also touch upon the background of my target audience.** (SEE Section 3.3)
Accessibility choices

2.3 Agile project management description

** Introduce Agile development as a concept and then discuss the details how how it was used. Benefits and drawbacks, as well as the importance of user feedback. Could talk about how user feedback was collected here but could also save that for the results and discussion section. Might be good to have it here though. **

Chapter 3

Results

3.1 Agile development

**Here I should discuss the success of the agile methodology in allowing me to perform rapid changes and meet user requirements better. How easy was it to make changes? How many changes were requested and what was the scale of those changes? Did users agree or disagree and how did I reconcile that? **

3.2 Evaluation of user feedback

3.2.1 User data

** Present data collected from users. Briefly discuss my impressions of how users found the course from their responses to the final questionnaire. **

3.2.2 Analysis

** Compare their responses with some resource I can find online. Maybe an empirical study has done something similar. As always, provide plenty of references. As a result of the comparison, evaluate what was successful and what was unsuccessful about the project. Could compare against more advanced online educational services (codecademy, brilliant, khan etc). **

Chapter 4

Discussion

4.1 Conclusions

** Did I meet the project goals set out at the start? Do I fully know if it was a success or a failure and why? What would I do differently if I were to do it again? **

4.2 Ideas for future work

** Possibility for using more sophisticated means of teaching. Deeper exploration of Finite Element Analysis, maybe room for a part 2 to the course? **

References

[1] Michael Renardy and Robert C. Rogers. *An Introduction to Partial Differential Equations*. Number 13 in Texts in Applied Mathematics. Springer=Verlag, New York, 2 edition, 2004.

[2] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Number 15 in Texts in Applied Mathematics. Springer Science + Business Media, New York, 2008.

[3] Mats G. Larson and Fredrik Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. Number 10 in Texts in Computational Science and Engineering. Springer-Verlag, New York, 2013.

[4] E. Toni. Variational formulation for every nonlinear problem. *International Journal of Engineering Science*, 22(11/12):1343–1371, 1984.

Appendix A

Self-appraisal

<This appendix should contain everything covered by the 'self-appraisal' criterion in the mark scheme. Although there is no length limit for this section, 2—4 pages will normally be sufficient. The format of this section is not prescribed, but you may like to organise your discussion into the following sections and subsections.>

A.1 Critical self-evaluation

A.2 Personal reflection and lessons learned

A.3 Legal, social, ethical and professional issues

<Refer to each of these issues in turn. If one or more is not relevant to your project, you should still explain *why* you think it was not relevant.>

A.3.1 Legal issues

A.3.2 Social issues

A.3.3 Ethical issues

A.3.4 Professional issues

Appendix B

External Material

<This appendix should provide a brief record of materials used in the solution that are not the student's own work. Such materials might be pieces of codes made available from a research group/company or from the internet, datasets prepared by external users or any preliminary materials/drafts/notes provided by a supervisor. It should be clear what was used as ready-made components and what was developed as part of the project. This appendix should be included even if no external materials were used, in which case a statement to that effect is all that is required.>