# Report for "Development of A Conceptualized Guided Coding For The Course of Mathematical Foundation of Finite Element Methods"

Xiaoming He

Assistant Professor

Department of Mathematics and Statistics

Office Phone: 573-341-6208

Email: hex@mst.edu

Project period: 07/01/2014-05/31/2015

***Summary***

Traditional finite element courses usually focus on either mathematical theory without enough implementation instructions or engineering applications with existing software. Meanwhile the capability to form new code packages or modify existing packages is critical for the career of many students in engineering and applied mathematics. However, this important bridge between the theory and applications is missed in the traditional finite element courses. This project develops a conceptualized guided coding, which is systematically constructed based on a thorough understanding of the concepts and general framework in the lectures so that the students can form their own code packages or modify existing packages under the general framework. In order to make the students fully understand the guided coding, we redesign or add the lectures, guided coding, homework assignments, exams, and independent study project one by one so that the students can follow these steps to gradually improve their understanding and capability. Benchmark problems are set up in the homework assignments and projects for the students to test their code packages, show how much they have improved, and find out what kind of difficulties they meet. As the pre-, mid- and post-assessments, three surveys for the students are designed to measure students' attitudes toward the course redesign for the guided coding and the improvement in their capability. The students' performance and feedback show that most of them understand the concepts and general framework in the lectures so that they can follow the guided coding to finish the basic code packages in the homework assignments, improve the basic packages for different problems in the midterm and final projects, and then extend the packages for more challenging problems in the independent study. This will get the students ready to form new code packages or modify existing packages with the general framework in their future works if they continue to practice for different problems.

***Purpose of project***

Finite element methods have been widely used for many different types of problems in computational science and engineering. This is why many universities have offered several different finite element courses at mathematics and engineering departments. However, the traditional finite element courses offered by a mathematics department usually focus on the mathematical theory without enough implementation instructions. On the other hand, the traditional finite element courses offered by an engineering department usually focus on the implementation and applications of finite element methods with existing software.

Therefore, the bridge between the theory and the applications, which is to form new finite element packages under a unified framework and modify existing packages for different purposes on their own, is missed in these traditional finite element courses, hence becomes our main purpose of this project. This bridge is a major requirement of the PhD program of computational and applied mathematics due to the needs to develop new algorithms. This bridge is also an important capability for the career of many engineering students since the existing software may not be able to handle all of their problems.

In this project, we develop a systematically conceptualized guided coding in a general framework to dynamically connect the concepts in the theoretical derivation and framework with the implementation issues in practice. This new guided coding is constructed based on a thorough understanding of all the related concepts and the corresponding implementation issues at each step of the construction of the finite element method so that the students are capable of forming their own code packages or modifying existing packages with the general framework. Therefore, as the bridge between the theory and applications, the conceptualized guided coding can enhance student professional development, increase faculty-student interaction, and promote active learning.

## ***Methodology***

In order to develop the conceptualized guided coding, all the major components of the course are systematically redesigned to advance the students' understanding and capability step by step. The lectures are first redesigned to deliver a thorough understanding from the original problems all the way to the corresponding algorithms and pseudo code. Then the guided coding is added to provide the instructions on the implementation and the general framework for transferring the pseudo codes and concepts in the lectures into real codes. The homework assignments are re-constructed and office hours are utilized to help students finishing a correct fundamental code package under the general framework. The midterm and final exams are two take-home projects which can improve the fundamental package to deal with different problems. Finally, the independent study project is added in order to extend the package for challenging problems, such as those in the students' research.

The lectures of the traditional finite element course offered by a mathematics department do not provide a straightforward view on the relationship between the theoretical concepts/derivations and their roles in the code since they focus on the theoretical and abstract point of view of the finite element methods and more advanced theoretical conclusions with their proofs. Therefore, the lectures for the theoretical concepts and derivations need to be first redesigned to fit the needs of the conceptualized guided coding. The major role of the redesigned the lectures and slides is to serve as a bridge between the original problems of partial differential equations and the pseudo code by delivering a thorough understanding of all the concepts and derivations at all the intermediate steps. In the lectures, the original partial differential equations are transferred into the pseudo code in the following five steps:

(1) Introduce the ***fundamental concepts and equations*** in the traditional mathematical way.
(2) Further explain the fundamental concepts by using ***more practical formulations*** with straightforward information for coding.
(3) Analyze the practical formulations to find the ***patterns and features*** for the implementation in the general framework.
(4) Illustrate how the ***data structures and subroutines*** are defined and used in the general coding framework by using examples, graphs, and tables.
(5) Describe the major algorithms used in the finite element methods in ***pseudo-code***.

One typical example, which provides more details about these five steps, can be found in the presentation for the Educational Research Symposium in March 2015.

Once the students fully understand the redesigned lectures, they already understand the key components and steps to code for the methods from the pseud-codes even though they have not started to code in a computer language. The pseud-code in the lectures can be directly copied into the real code as a basic code and then enriched for different implementation details. These redesigned lectures are critical to make the coding work much easier and better organized for all the later stages, including the guided coding, homework assignments, midterm and final projects, and independent study.

The major role of the guided coding is to serve as a bridge between the pseudo-code and the real code with all implementation issues in detail. The students start from the design of a general structure of the package based on the general framework and key features of finite element methods which are discussed in the lecture slides. Then the major subroutines will be constructed for the core parts of the algorithm in slides and partially finished under the general framework. A group of parameters and data structures need to be designed in order to explicitly reflect the concepts in a one-to-one correspondence way. Since the students need both instructions and practice on the design and coding, some works are done in the guided coding together with the students and the others are left to the students for stimulating them to think and practice. Each concept or function in the theoretical derivation of the algorithm has an explicit representation in the code, such as a subroutine, a parameter, a loop, a judgement, a row of code, and so on. Once the package is completed, the logic relationship between the theoretical structures, concepts, functions of finite elements and the practical implementation issues of the algorithm are very clear to the students. All the steps of the guided coding will be featured by a constant and dynamic interaction between the instructor and the students in the open discussion and question-answer style.

In homework assignments, instead of many theoretical or engineering application questions in the traditional finite element courses, the students are asked to practice on the coding-related concepts in the re-designed lectures, use these concepts to complete the basic code package from the guided coding, and test the package by using the different benchmark problems. The benchmark problems are some standard numerical experiments with analytic solutions for different types of problem domains and boundary conditions. Once the students complete the code package to obtain the numerical solutions, the errors between the numerical solutions and the analytic solutions can be computed. These errors can mathematically demonstrate the accuracy of the students' code packages. If a student cannot completely figure out the solution and the code before it's graded, the student can use the solution code to find out the mistakes and then correct them. The modification of the students' code packages for different types of problem domains and boundary conditions shows us whether the code packages can be easily modified for different cases. These re-designed homework assignments will stimulate the students to think about and try more for the guided coding and related concepts, hence help them improving and completing their understanding of the lectures and guided coding through their questions during office hours. The total points for the homework assignments are 100 to grade on the students' basic understanding of the lectures and guided coding.

For the midterm and final exams, two take-home projects are used to replace the in-class exams. The projects will test the students about a more complete understanding of all the concepts and implementation issues as part of the whole general framework, not just some individual aspects tested by individual questions. The projects will further improve the students' capability to form their own code packages since they need to upgrade the basic packages from the homework assignments to more advanced ones. The target problems in the projects are based on those in the homework assignments but more complicated and advanced. Before the students start working on the projects, they learn different methods and techniques for extending the major components of finite element methods in lectures. Through a short guided coding, the students also learn about the major ideas for the corresponding implementation, but not as much details as in the regular guided coding for the basic packages of the homework assignments. Then they need to think through the general framework and specific designs by themselves for the implementation based on what they have learned. With all of these preparations, they can finish the packages and test them by using some given benchmark problems. The midterm project has 100 points and the final project has 200 points to exam the students for their complete understanding of the materials and general framework, and their implementation capability.

In order to provide an opportunity for the students to challenge themselves, an independent study project is added in the last month of the semester. It is also an opportunity for the students to tailor this course to satisfy their specific interests in computational science and engineering. Based on what has been taught in class, each student needs to choose a challenging topic which is interesting to the student. After a student submits a written description of the proposed topic and the plan for carrying out this study, individual discussion is utilized to improve and finalize the topic and plan. The project report should include both the theoretical and the practical parts of the independent study. After the background of the problem is introduced and the formulations of the method are derived for the target problem, the report needs to discuss the details to design, finish, and test the corresponding code package. In order to grade on the students' capability improvement for applying and extending what they have learned in class to their future works, the 100 points for the independent study include three parts: 30 points for the difficulty and selection procedure of the proposed topic, 30 points for the degree of completion of the plan, and 40 points for the results.

If the students can understand the main concepts for the given partial differential equations, they should be able to quickly find the corresponding components in the code package and modify them correspondingly since the theoretical concepts and the code components are one-to-one corresponding to each other in our conceptualized guided coding. Therefore, in all of the homework assignments and projects, the students are evaluated for how many concepts and the corresponding components in the packages they can correctly identify and understand for the given partial differential equations, how many new problems they can successfully modify the package for, how fast they can finish the modification for each problem, and whether they can obtain the optimal accuracy order in L2 norm, H1 norm, and infinity norms. These measurements can illustrate the improvement of the students' understanding of the theoretical concepts for the coding and the capability of the students to modify their own packages for different problems.

If the students can correctly form their own code packages through the conceptualized guided coding, modify their packages for different problems in homework assignments and projects, and successfully solve all the benchmark problems, then it can be concluded that the students gain a thorough understanding of all the concepts and implementation issues of finite element methods as well as the capability to form their own code packages or modify existing packages. This is one major criterion to decide if the course redesign for the conceptualized guided coding is successful or not. It is also an important indication for the significant enhancement of students' conceptual understanding.

As the pre-, mid- and post-assessments, three surveys for the students are also designed to measure students' attitudes toward the course redesign for the guided coding and the improvement in their capability. In order to see what the students can gain in this course, the pre-assessment investigates on the background of the students in both the finite element methods and the general numerical/coding techniques. It also collects the expectations of the students for different aspects of this course. The mid- and post-assessments investigate on how deep the students have understood the finite element methods as well as how helpful the re-designed and added course components are for improving the students' understanding and capability of both the finite element methods and the general numerical/coding techniques. The mid-assessment also asks students whether their expectations for this course have changed or not in the middle of the semester. The post-assessment also asks the students whether their expectations have been reached or not at the end of the semester.

***Results***

17 students finished all the study for the whole semester, including two students from University of Missouri at Columbia through the course sharing program. Most of them understood the theoretical framework and concepts in both the traditional and practical ways, the roles which the theoretical framework and concepts play in the coding procedure, and the techniques to translate them into different components of the code under the general framework. Hence they obtained a thorough understanding of finite element methods, a lot of experience in coding, their own finite element packages under the general framework as well as the capability to understand the existing code packages and modify them for different purposes.

Considering the homework assignments, the students' performance was gradually improved for more and more complicated problems while they learn more and more from the newly designed lectures and the guided coding. For the 1$^{st}$ homework assignment which is for a basic 1D problem, 13 students successfully figured out the correct results and code package for the benchmark problems. The other students also well understood the solutions and code after they were distributed to the students. For the 2$^{nd}$ homework assignment which is an extension of the basic 1D problem, 15 students successfully obtained the correct results and code package. For the 3$^{rd}$ homework assignment which is a relatively easier 2D work as an extension of the preparation work in 1D to 2D, 15 students successfully obtained the correct results and subroutines. For the 4$^{th}$ homework assignment which is more difficult than the previous ones since it is the first full 2D problem, 12 students successfully obtained the correct results and code package. For the 5$^{th}$ homework assignment which is even more complicated than the 4$^{th}$ one, 13 students successfully obtained the correct results and code package. The other students also finished most of the homework assignments but were not able to completely figure them out due to some mistakes or misunderstanding which were corrected after the solutions were distributed to the students.

After the preparation in the 1st and 2nd homework assignments, the students took the midterm project to complete the 1D code package and apply it for different numerical tests. 16 students successfully finished this project. Only one student did not completely finish the package but the student's package still passed half of the numerical tests. After the project was graded, the student found out the problem and successfully passed all the numerical tests. After the preparation in the 3rd, 4th, and 5th homework assignments as well as some further preparation in the lectures and guided coding, the students took the final project for an unsteady 2D problem. 13 students successfully finished this project. The other 4 students, who made some mistakes in the final project, actually performed better than the 4th and 5th homework assignments even though they do need further study and practice for what they misunderstood.

In the independent study, most of the students chose very challenging problems and well made use of what they have learned in class to at least partially resolve those hard problems. Several students did a very good job to beautifully illustrate how the general package can be easily modified for different difficult realistic problems based on a thorough understanding of the difference between the new problems and the course. In their reports, most of the students well described their target problems, introduced the background, derived the numerical algorithm, carried out the numerical experiments, and showed the numerical results. This indicates that they well understand the correct way to connect what they have learned in class with what they may need to do in their works.

From the pre-assessment, we can see that only several students learned about the finite element methods through a course or by themselves to partially understand the core components of the finite element methods or the general framework which are used by most of the existing software and packages. None of them fully understood the finite element methods, formed their own finite element package, or modified an existing finite element package. Only about half of the students took other numerical courses before or had coding experience. Only one student had extensive coding experience. Most of the students expected to obtain a complete understanding of the basics of the finite element methods, some extensions to more general and complicated problems, the capability to form or modify a finite element package, and improvements in their coding techniques.

Based on the students' backgrounds, it is not straightforward to reach their expectations in one semester, especially for those who are lack of backgrounds in numerical methods or coding. However, the students' performance and their feedback in the later assessments show that the redesigned course with the guided coding did a good job to reach the expectations of most of the students.

From the mid- and post-assessments, most of the students believe

(1) the lecture slides provide enough information on the concepts, structures, derivations, and understanding for the conceptualized guided coding;
(2) the conceptualized guided coding provides enough instructions for the homework, project, and their understanding about the implementation of the finite element method;
(3) the homework assignments and projects stimulate them enough to think through and finish the code which the lecture slides and conceptualized guided coding discuss about but do not finish;

This indicates that the redesign of the major course components was successful. Most of the students believe they understood the core components and the general coding framework of the finite element method. But about half of the students only partially understood it, which needs further investigation for the future improvement of the course. Most of the students believe that they can (at least partially) form their own finite element package or modify an existing package. Most of the students believe that this course improved their capability for studying and coding other numerical methods as well as their general coding techniques. Most of the students did not change their initial expectations for this course in the mid-assessment. About two thirds of the students believed that their expectations were reached and about one third of the students believed that their expectations were partially reached.

## ***Conclusions***

Based on the students' performance and feedback, we successfully used the conceptualized guided coding and the correspondingly redesigned course to reach our project target, which is to deliver students the bridge between the theory and applications of the finite element methods. The students gain a thorough understanding of the methods and the capability to form new finite element packages under the general framework for different purposes on their own. Since this unified framework is a key feature of the finite element methods for their practical use, most of the existing packages and software use the same framework. Hence the students will be also able to understand the existing packages and modify them for different purposes. Many students also significantly improve their coding techniques through this course. With more practice for the students' packages and the existing packages under the general framework, most of the students will be able to extend their own packages and modify existing packages for their future works.

In the future, it will be important to further redesign/refine the homework assignments and projects, and provide more instructions on independent study in order to deliver a better understanding and fully take advantages of what the students have learned in the lectures and guided coding. In order to cover a wider range of students, it will be also helpful to provide more preliminary materials for the students who are lack of enough background in numerical methods as well as more theoretical materials for the students who are more interested in the theory.