Development of a Conceptualized Guided Coding for Finite Element Methods

Xiaoming He
Department of Mathematics & Statistics
Missouri University of Science & Technology

Supported by Educational Research grant at Missouri S&T

Outline

Outline

Background

- Finite element methods: very important and fundamental numerical methods for solving partial differential equations of different problems in computational science and engineering.

Background

- Finite element methods: very important and fundamental numerical methods for solving partial differential equations of different problems in computational science and engineering.
- Many schools offer several different finite element courses in mathematics and engineering departments

Background

- Finite element methods: very important and fundamental numerical methods for solving partial differential equations of different problems in computational science and engineering.
- Many schools offer several different finite element courses in mathematics and engineering departments
- In mathematics departments: mathematical theory without enough implementation instructions

Background

- Finite element methods: very important and fundamental numerical methods for solving partial differential equations of different problems in computational science and engineering.
- Many schools offer several different finite element courses in mathematics and engineering departments
- In mathematics departments: mathematical theory without enough implementation instructions
- In engineering departments: engineering applications with existing software

Motivation

- What is missed between these two types of finite element courses?

Motivation

- What is missed between these two types of finite element courses?
- The capability to self-implement finite element methods to independently form a new finite element package under a general framework based on a thorough understanding of finite element methods.

Motivation

- What is missed between these two types of finite element courses?
- The capability to self-implement finite element methods to independently form a new finite element package under a general framework based on a thorough understanding of finite element methods.
- Or the capability to correctly and quickly understand and modify open-source packages for new problems.

Motivation

- What is missed between these two types of finite element courses?
- The capability to self-implement finite element methods to independently form a new finite element package under a general framework based on a thorough understanding of finite element methods.
- Or the capability to correctly and quickly understand and modify open-source packages for new problems.
- This is one of the main requirements of the Ph.D program of computational and applied mathematics

Motivation

- What is missed between these two types of finite element courses?
- The capability to self-implement finite element methods to independently form a new finite element package under a general framework based on a thorough understanding of finite element methods.
- Or the capability to correctly and quickly understand and modify open-source packages for new problems.
- This is one of the main requirements of the Ph.D program of computational and applied mathematics
- This is also an critical capability for the career of students in other majors since the existing software may not be able to handle all of their problems.

Proposed work

- How to make up this bridge between the math and engineering courses on finite elements?

Proposed work

- How to make up this bridge between the math and engineering courses on finite elements?
- 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.

Expectations

The conceptualized guided coding can

- Enhance student professional development
- Increase faculty-student interaction
- Promote active learning
- ......

Outline

Course structure

- Redesigned lectures for a thorough understanding of the methods

Course structure

- Redesigned lectures for a thorough understanding of the methods
- Conceptualized guided coding for implementation instructions

Course structure

- Redesigned lectures for a thorough understanding of the methods
- Conceptualized guided coding for implementation instructions
- Homework assignments and office hour to complete the fundamental package

Course structure

- Redesigned lectures for a thorough understanding of the methods
- Conceptualized guided coding for implementation instructions
- Homework assignments and office hour to complete the fundamental package
- Midterm and final projects to improve the package to deal with different problems.

Course structure

- Redesigned lectures for a thorough understanding of the methods
- Conceptualized guided coding for implementation instructions
- Homework assignments and office hour to complete the fundamental package
- Midterm and final projects to improve the package to deal with different problems.
- Independent study project to extend the package for challenging problems.

Redesigned lectures

- Major task: serve as the bridge between the partial differential equations and the pseudo-code.

Redesigned lectures

- Major task: serve as the bridge between the partial differential equations and the pseudo-code.
- Introduce the fundamental concepts and equations in the traditional mathematical way.

Redesigned lectures

- Major task: serve as the bridge between the partial differential equations and the pseudo-code.
- Introduce the fundamental concepts and equations in the traditional mathematical way.
- Further explain the fundamental concepts by using more practical formulations with straightforward information for coding.

Redesigned lectures

- Major task: serve as the bridge between the partial differential equations and the pseudo-code.
- Introduce the fundamental concepts and equations in the traditional mathematical way.
- Further explain the fundamental concepts by using more practical formulations with straightforward information for coding.
- Analyze the practical formulations to find the patterns and features for the implementation in the general framework.

Redesigned lectures

- Major task: serve as the bridge between the partial differential equations and the pseudo-code.
- Introduce the fundamental concepts and equations in the traditional mathematical way.
- Further explain the fundamental concepts by using more practical formulations with straightforward information for coding.
- Analyze the practical formulations to find the patterns and features for the implementation in the general framework.
- Illustrate how the data structures and subroutines are defined and used in the general coding framework by using examples, graphs, and tables.

Redesigned lectures

- Major task: serve as the bridge between the partial differential equations and the pseudo-code.
- Introduce the fundamental concepts and equations in the traditional mathematical way.
- Further explain the fundamental concepts by using more practical formulations with straightforward information for coding.
- Analyze the practical formulations to find the patterns and features for the implementation in the general framework.
- Illustrate how the data structures and subroutines are defined and used in the general coding framework by using examples, graphs, and tables.
- Describe the major algorithms used in the finite element methods in pseudo-code.

Conceptualized guided coding

- Major task: serve as the bridge between the pseudo-code and the real code.

Conceptualized guided coding

- Major task: serve as the bridge between the pseudo-code and the real code.
- 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.

Conceptualized guided coding

- Major task: serve as the bridge between the pseudo-code and the real code.
- 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.
- The major subroutines will be constructed for the core parts of the algorithm derived theoretically in slides and partially finished under the general framework.

Conceptualized guided coding

- Major task: serve as the bridge between the pseudo-code and the real code.

- 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.

- The major subroutines will be constructed for the core parts of the algorithm derived theoretically in slides and partially finished under the general framework.

- A group of parameters and data structures will be designed in order to illustrate how to explicitly reflect the concepts in a one-to-one correspondence way. Others will be left to the students for their thinking about the design in a similar way.

Conceptualized guided coding

- Each concept or function in the theoretical derivation of the algorithm will have an explicit representation in the code.

Conceptualized guided coding

- Each concept or function in the theoretical derivation of the algorithm will have an explicit representation in the code.
- 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 will be very clear to the students.

Conceptualized guided coding

- Each concept or function in the theoretical derivation of the algorithm will have an explicit representation in the code.
- 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 will be 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.

Homework and office hours

- Practice on the concepts learned in lecture.

Homework and office hours

- Practice on the concepts learned in lecture.
- Complete the code package

Homework and office hours

- Practice on the concepts learned in lecture.
- Complete the code package
- Test the package by using the given numerical examples.

Homework and office hours

- Practice on the concepts learned in lecture.
- Complete the code package
- Test the package by using the given numerical examples.
- Further improve and complete the students' understanding of the lectures and guided coding by answering questions.

Midterm and final projects

- Learn different choices for different major components of finite element methods in lectures.

Midterm and final projects

- Learn different choices for different major components of finite element methods in lectures.
- Briefly learn about the major ideas for the corresponding implementation through a short guided coding.

Midterm and final projects

- Learn different choices for different major components of finite element methods in lectures.
- Briefly learn about the major ideas for the corresponding implementation through a short guided coding.
- Think through the general structure and specific designs for the implementation.

Midterm and final projects

- Learn different choices for different major components of finite element methods in lectures.
- Briefly learn about the major ideas for the corresponding implementation through a short guided coding.
- Think through the general structure and specific designs for the implementation.
- Finish the package and test it by using the given numerical examples.

Independent study

- Choose an interesting and challenging topic.

Independent study

- Choose an interesting and challenging topic.
- A written description of the proposed topic and the plan for carrying out this study.

Independent study

- Choose an interesting and challenging topic.
- A written description of the proposed topic and the plan for carrying out this study.
- Individual discussion for the topic and plan.

Independent study

- Choose an interesting and challenging topic.
- A written description of the proposed topic and the plan for carrying out this study.
- Individual discussion for the topic and plan.
- Report for the theoretical part of the independent study.

Independent study

- Choose an interesting and challenging topic.
- A written description of the proposed topic and the plan for carrying out this study.
- Individual discussion for the topic and plan.
- Report for the theoretical part of the independent study.
- Design, finish, and test the corresponding package to complete the final report.

Independent study

- Choose an interesting and challenging topic.
- A written description of the proposed topic and the plan for carrying out this study.
- Individual discussion for the topic and plan.
- Report for the theoretical part of the independent study.
- Design, finish, and test the corresponding package to complete the final report.
- This is students' opportunity to tailor this course to satisfy their specific interests in numerical analysis.

Learning outcomes

The students will learn or obtain

- theoretical framework and concepts in both the traditional and a practical ways;

Learning outcomes

The students will learn or obtain

- theoretical framework and concepts in both the traditional and a practical ways;
- the roles which the theoretical framework and concepts play in the coding procedure and how to translate them into different components of the code under a general coding framework;

Learning outcomes

The students will learn or obtain

- theoretical framework and concepts in both the traditional and a practical ways;
- the roles which the theoretical framework and concepts play in the coding procedure and how to translate them into different components of the code under a general coding framework;
- a thorough understanding of finite element methods;

Learning outcomes

The students will learn or obtain

- theoretical framework and concepts in both the traditional and a practical ways;
- the roles which the theoretical framework and concepts play in the coding procedure and how to translate them into different components of the code under a general coding framework;
- a thorough understanding of finite element methods;
- their own finite element packages under a unified framework;

Learning outcomes

The students will learn or obtain

- theoretical framework and concepts in both the traditional and a practical ways;
- the roles which the theoretical framework and concepts play in the coding procedure and how to translate them into different components of the code under a general coding framework;
- a thorough understanding of finite element methods;
- their own finite element packages under a unified framework;
- capability to understand the existing open-source code packages and modify them for different purposes.

An example for the redesigned lectures

- Consider the 2D second order elliptic equation

$$-\nabla \cdot (c\nabla u) = f, \quad \text{in } \Omega$$
$$u = g, \quad \text{on } \partial\Omega.$$

  where Ω is a 2D domain, $f(x,y)$ and $c(x,y)$ are given functions on Ω, $g(x,y)$ is a given function on $\partial\Omega$ and $u(x,y)$ is the unknown function.

- The gradient of a 2D function u is defined by

$$\nabla u = (u_x, u_y).$$

- The divergence of a 2×1 vector $\overrightarrow{v}$ is defined by

$$\nabla \cdot \overrightarrow{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y}.$$

- ......

An example for the redesigned lectures

- ......
- The weak formulation: find $u \in H^1(\Omega)$ such that

$$\int_\Omega c\nabla u \cdot \nabla v \; dxdy = \int_\Omega fv \; dxdy.$$

  for any $v \in H_0^1(\Omega)$.
- ......

An example for the redesigned lectures

- ......
- The Galerkin formulation is to find $u_h \in U_h$ such that

$$a(u_h, v_h) = (f, v_h)$$
$$\Leftrightarrow \quad \int_\Omega c \nabla u_h \cdot \nabla v_h \, dxdy = \int_\Omega f v_h \, dxdy$$

  for any $v_h \in U_h$.
- Here $U_h = span\{\phi_j\}_{j=1}^{N_b}$ is chosen to be a finite element space where $\{\phi_j\}_{j=1}^{N_b}$ are the global finite element basis functions.
- ......

An example for the redesigned lectures

- ......
- Since $u_h \in U_h = span\{\phi_j\}_{j=1}^{N_b}$, then

$$u_h = \sum_{j=1}^{N_b} u_j \phi_j$$

for some coefficients u_j $(j = 1, \cdots, N_b)$.

- Choose the test function $v_h = \phi_i$ $(i = 1, \cdots, N_b)$
- The discretization formulation gives

$$\int_\Omega c \nabla \left(\sum_{j=1}^{N_b} u_j \phi_j \right) \cdot \nabla \phi_i \ dxdy = \int_\Omega f \phi_i \ dxdy,$$

$$\Rightarrow \sum_{j=1}^{N_b} u_j \left[\int_\Omega c \nabla \phi_j \cdot \nabla \phi_i \ dxdy \right] = \int_\Omega f \phi_i \ dxdy, \ i = 1, \cdots, N_b.$$

- ......

An example for the redesigned lectures

- ......
- Define the stiffness matrix

$$A = [a_{ij}]_{i,j=1}^{N_b} = \left[\int_{\Omega} c \nabla \phi_j \cdot \nabla \phi_i \; dxdy \right]_{i,j=1}^{N_b}.$$

- Define the load vector

$$\vec{b} = [b_i]_{i=1}^{N_b} = \left[\int_{\Omega} f \phi_i \; dxdy \right]_{i=1}^{N_b}.$$

- Define the unknown vector

$$\vec{X} = [u_j]_{j=1}^{N_b}.$$

- Then we obtain the matrix formulation

$$A\vec{X} = \vec{b}.$$

- ......

An example for the redesigned lectures

- ......
- From the definition of ϕ_j $(j = 1, \cdots, N_b)$, we can see that ϕ_j are non-zero only on the elements adjacent to the node X_j, but 0 on all the other elements.
- This observation motivates us to think about

$$a_{ij} = \int_\Omega c \nabla \phi_j \cdot \nabla \phi_i \, dxdy = \sum_{n=1}^{N} \int_{E_n} c \nabla \phi_j \cdot \nabla \phi_i \, dxdy.$$

- It is easy to see that most of $\int_{E_n} c \nabla \phi_j \cdot \nabla \phi_i \, dxdy$ will be 0.
- So we only need to use numerical integration to compute those nonzero integrals.
- ......

An example for the redesigned lectures

General local assembly idea for A:

- Loop over all the elements;
- Compute all non-zero local integrals on each element for A;
- Assemble these non-zero local integrals into the corresponding entries of the stiffness matrix A.

An example for the redesigned lectures

- ......
- In fact, for $n = 1, \cdots, N$,

$$\int_{E_n} c\nabla\psi_{n\alpha} \cdot \nabla\psi_{n\beta}\, dxdy \ (\alpha, \beta = 1, \cdots, N_{lb})$$

should be assembled to a_{ij} where $i = T_b(\beta, n)$ and $j = T_b(\alpha, n)$.

- ......

An example for the redesigned lectures

Algorithm I-1:

- Initialize the matrix: $A = sparse(N_b, N_b)$;
- Compute the integrals and assemble them into A:

FOR $n = 1, \cdots, N$:

    FOR $\alpha = 1, \cdots, N_{lb}$:

        FOR $\beta = 1, \cdots, N_{lb}$:

            Compute $r = \int_{E_n} c \nabla \psi_{n\alpha} \cdot \nabla \psi_{n\beta} \, dxdy$;

            Add r to $A(T_b(\beta, n), T_b(\alpha, n))$.

        END

    END

END

An example for the redesigned lectures

To make a general subroutine for different cases, more information needed for computing and assembling the integral should be treated as input parameters or input functions of this subroutine:

- the coefficient function c;
- the quadrature points and weights for numerical integrals;
- the mesh information matrices P and T, which can also provide the number of mesh elements $N = size(T, 2)$ and the number of mesh nodes $N_m = size(P, 2)$;
- the finite element information matrices P_b and T_b for the trial and test functions respectively, which can also provide the number of local basis functions $N_{lb} = size(T_b, 1)$ and the number of the global basis functions $N_b = size(P_b, 2)$ ($=$ the number of unknowns);
- the type of the basis function for the trial and test functions respectively;

An example for the redesigned lectures

- Note that

$$\int_{E_n} c\nabla\psi_{n\alpha}\cdot\nabla\psi_{n\beta}\, dxdy = \int_{E_n} c\frac{\partial\psi_{n\alpha}}{\partial x}\frac{\partial\psi_{n\beta}}{\partial x}\, dxdy + \int_{E_n} c\frac{\partial\psi_{n\alpha}}{\partial y}\frac{\partial\psi_{n\beta}}{\partial y}\, dxdy.$$

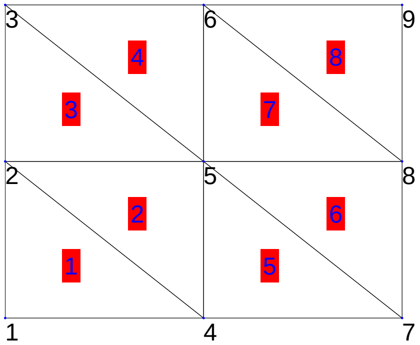
- Hence we can consider to develop an algorithm to assemble the matrix arising from a more general integral

$$\int_{E_n} c\frac{\partial^{r+s}\psi_{n\alpha}}{\partial x^r\partial y^s}\frac{\partial^{p+q}\psi_{n\beta}}{\partial x^p\partial y^q}\, dxdy.$$

with parameters r, s, p, and q.

An example for the redesigned lectures

- ......
- Define your global indices for all the mesh elements E_n $(n = 1, \cdots, N)$ and mesh nodes Z_k $(k = 1, \cdots, N_m)$.
- For example, when $N_1 = N_2 = 2$, we have



An example for the redesigned lectures

- ......
- Define matrix P to be an information matrix consisting of the coordinates of all mesh nodes.
- Define matrix T to be an information matrix consisting of the global node indices of the mesh nodes of all the mesh elements.
- For example, when $N_1 = N_2 = 2$, we have

$$P = \begin{pmatrix} 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 1 & 1 & 1 \\ 0 & 0.5 & 1 & 0 & 0.5 & 1 & 0 & 0.5 & 1 \end{pmatrix},$$

$$T = \begin{pmatrix} 1 & 2 & 2 & 3 & 4 & 5 & 5 & 6 \\ 4 & 4 & 5 & 5 & 7 & 7 & 8 & 8 \\ 2 & 5 & 3 & 6 & 5 & 8 & 6 & 9 \end{pmatrix}.$$

- ......

An example for the redesigned lectures

- ......
- The "reference→ local → global" framework is used to construct the finite element spaces. We consider the reference 2D linear basis functions on $\hat{E} = \triangle \hat{A}_1 \hat{A}_2 \hat{A}_3$ where $\hat{A}_1 = (0,0)$, $\hat{A}_2 = (1,0)$, and $\hat{A}_3 = (0,1)$.
- Define three reference 2D linear basis functions

$$\hat{\psi}_j(\hat{x},\hat{y}) = a_j\hat{x} + b_j\hat{y} + c_j, \ j = 1,2,3,$$

such that

$$\hat{\psi}_j(\hat{A}_i) = \delta_{ij} = \begin{cases} 0, & \text{if } j \neq i, \\ 1, & \text{if } j = i, \end{cases}$$

for $i,j = 1,2,3$.
- Then the three reference 2D linear basis functions are

$$\hat{\psi}_1(\hat{x},\hat{y}) = -\hat{x} - \hat{y} + 1, \ \hat{\psi}_2(\hat{x},\hat{y}) = \hat{x}, \ \hat{\psi}_3(\hat{x},\hat{y}) = \hat{y}.$$

- ......

An example for the redesigned lectures

- ......
- Now we can use the affine mapping between an arbitrary triangle $E = \triangle A_1 A_2 A_3$ and the reference triangle $\hat{E} = \triangle \hat{A}_1 \hat{A}_2 \hat{A}_3$ to construct the local basis functions from the reference ones.
- Assume

$$A_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \quad i = 1, 2, 3.$$

- Consider the affine mapping

$$\begin{aligned} \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} A_2 - A_1, A_3 - A_1 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + A_1 \\ &= \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \end{aligned}$$

- ......

An example for the redesigned lectures

- ......
- Local basis functions

$$\begin{aligned}
\psi_{n1}(x,y) &= \hat{\psi}_1(\hat{x},\hat{y}) = -\hat{x}-\hat{y}+1\\
&= -\frac{(y_{n3}-y_{n1})(x-x_{n1})-(x_{n3}-x_{n1})(y-y_{n1})}{|J_n|}\\
&\quad -\frac{-(y_{n2}-y_{n1})(x-x_{n1})+(x_{n2}-x_{n1})(y-y_{n1})}{|J_n|}+1,\\
\psi_{n2}(x,y) &= \hat{\psi}_2(\hat{x},\hat{y}) = \hat{x}\\
&= \frac{(y_{n3}-y_{n1})(x-x_{n1})-(x_{n3}-x_{n1})(y-y_{n1})}{|J_n|},\\
\psi_{n3}(x,y) &= \hat{\psi}_3(\hat{x},\hat{y}) = \hat{y}\\
&= \frac{-(y_{n2}-y_{n1})(x-x_{n1})+(x_{n2}-x_{n1})(y-y_{n1})}{|J_n|}.
\end{aligned}$$

- ......

Outline

1. Introduction
2. Course design
3. Evaluation and feedback
4. Conclusions

Plan for evaluation

The students will be evaluated for

- how many concepts and the corresponding components in the packages they can correctly identify and understand for the needs of the new partial differential equations

Plan for evaluation

The students will be evaluated for

- how many concepts and the corresponding components in the packages they can correctly identify and understand for the needs of the new partial differential equations
- how many new problems they can successfully modify the package for

Plan for evaluation

The students will be evaluated for

- how many concepts and the corresponding components in the packages they can correctly identify and understand for the needs of the new partial differential equations
- how many new problems they can successfully modify the package for
- how fast they can finish the modification for each problem

Plan for evaluation

The students will be evaluated for

- how many concepts and the corresponding components in the packages they can correctly identify and understand for the needs of the new partial differential equations
- how many new problems they can successfully modify the package for
- how fast they can finish the modification for each problem
- whether they can obtain the optimal accuracy order in L2 norm, H1 norm, and infinity norms

Plan for evaluation

Furthermore,

- Pre-, mid-, and post-assessments will be designed to measure students' attitudes and their capability and self-efficacy.

Plan for evaluation

Furthermore,

- Pre-, mid-, and post-assessments will be designed to measure students' attitudes and their capability and self-efficacy.
- In the teaching evaluation, several questions will be also asked by the instructor to evaluate the students' satisfaction for what the students learn from the conceptualized guided coding versus the theory the students can learn from the traditional theoretical finite element course.

Pre-assessment

- 10 students in math
- 2 students in physics
- 2 students in computer science
- 2 students in aerospace engineering
- 1 students in nuclear engineering
- 1 students in petroleum engineering

Pre-assessment

- Only 1 student took another finite element course: AE 5212;

Pre-assessment

- Only 1 student took another finite element course: AE 5212;
- Only 2 students learned about how to derive and construct the finite element method; only 2 students partially learned about it;

Pre-assessment

- Only 1 student took another finite element course: AE 5212;
- Only 2 students learned about how to derive and construct the finite element method; only 2 students partially learned about it;
- No student learned about how to define and construct the finite element basis functions; only 3 students partially learned about it;

Pre-assessment

- Only 1 student took another finite element course: AE 5212;
- Only 2 students learned about how to derive and construct the finite element method; only 2 students partially learned about it;
- No student learned about how to define and construct the finite element basis functions; only 3 students partially learned about it;
- Only 1 student learned about the general framework for implementing finite element method, which is used by most of the existing finite element software and open-source finite element packages; only 2 students partially learned about it;

Pre-assessment

- Only 1 student learned about the local assembly procedure (computing all the matrix entries on local elements and then assembling them to the global stiffness matrix) for finite element methods; only 2 students partially learned about it;

Pre-assessment

- Only 1 student learned about the local assembly procedure (computing all the matrix entries on local elements and then assembling them to the global stiffness matrix) for finite element methods; only 2 students partially learned about it;

- No student successfully coded for a finite element method on your own without using any existing finite element software or package; only 1 student partially learned about it;

Pre-assessment

- Only 1 student learned about the local assembly procedure (computing all the matrix entries on local elements and then assembling them to the global stiffness matrix) for finite element methods; only 2 students partially learned about it;
- No student successfully coded for a finite element method on your own without using any existing finite element software or package; only 1 student partially learned about it;
- No student successfully understood or modified any open-source finite element package; only 2 students partially learned about it;

Pre-assessment

- 9 students took or are taking at least one course course for numerical methods in addition to this course: Numerical Analysis, Numerical differential equations, Numerical Linear Algebra, Numerical Meththods;

Pre-assessment

- 9 students took or are taking at least one course course for numerical methods in addition to this course: Numerical Analysis, Numerical differential equations, Numerical Linear Algebra, Numerical Meththods;
- 9 students students had coding experience: 0.5-4 years; C, C++, Fortran, Matlab, Java, Mathematica......; 15-5000 lines of code;

Pre-assessment

- 9 students took or are taking at least one course course for numerical methods in addition to this course: Numerical Analysis, Numerical differential equations, Numerical Linear Algebra, Numerical Meththods;
- 9 students students had coding experience: 0.5-4 years; C, C++, Fortran, Matlab, Java, Mathematica......; 15-5000 lines of code;
- Only 2 students successfully modified at least one existing package for some numerical methods: 30 or 1000 lines of code;

Pre-assessment

Considering Math 6602 as your first finite element course in math, what is your expectation from this course for the level of understanding of the finite element methods and the difficulty to reach this level?

- I expect fairly simple, more general examples in class and the guided coding of a difficulty that would be moderate to high depending on my previous understanding of the material (both the mathematics and the programming).
- I expect to be able to understand all of the underlying math of FE techniques and to be able to very quickly modify existing FE software packages.

Pre-assessment

- I expect to learn a very general framework of coding and have the opportunity to practice with it, familiarizing myself with its flexibility and function. Difficulty level is relative.
- to learn and understand the finite element method and use it in the future research
- Up to being able to solve simple multidimensional time-varying problems with FEMs. I think this is achievable in a one semester course.
- Start by a mathematical introduction to the FE and then do a practice on coding to improve ourself until we can do new coding individually.

Pre-assessment

- I expect that I can understand this course very well some times some parts of the slide are not very clear and if you provide an example mathematically to explain I think it could be help us to dominate the difficulty.
- after the class, i can code any easy function by mysely
- Just understand how to use this to solve the basic problems.
- I expect to understand the basics of the finite element method and how to translate those basics into code. I expect this to be moderately difficult given my poor understanding of the basic themselves and lack of experience with MATLAB and other numerical coding applications.

Pre-assessment

Considering Math 6602 as your first finite element course in math, what is your expectation from this course for improving your understanding about how to solve different partial differential equations by using finite element methods?

- I expect to learn the general solution to simpler systems, with a fairly recognizable path for extrapolating the method for more complicated systems.
- I expect to be able to solve any k-dimensional transient pde using finite element methods after this course (or at least be able to extend the code we develop to meet such needs).
- to learn how to solve differential equations efficiently
- If I could know enough to expand into n-dimensions and come up with simple variations on the topics discussed in class, I will be satisfied.

Pre-assessment

- My expectation was to understand mathematical principles [preceding the formulation of the computational methods] to a degree that was general enough to offer a wide range of understanding in how to solve PDEs with FEMs. In essence, take any PDE (of certain favorable properties) and know a general algorithm for preparing it for FEMs.
- Problems of partial differential equations should be explained first and then do an extension.
- I expect we can learn concepts to create coding and expansion the codes to solve several kind of PDE.

Pre-assessment

The guided coding in class will lead all students to obtain a fundamental package after class. What is your expectation from Math 6602 for improving your capability to apply your own fundamental package to different types of partial differential equations with different levels of difficulty? (Or, what types of equations do you expect to be able to solve by using your fundamental package?)

- Seeing as I have no real prior experience in programming, I expect that I will be able to usev the package in conjunction with MatLab, and will be able to solve simpler systems of undetermined degree, if my understanding of the sparse function is correct.
- Though I may be distant from this goal now, I hope to extrapolate my understanding to approximating solutions to the neutron transport equation

Pre-assessment

- I expect that with little modification it will be able to solve any k-dimensional transient pde.
- Right now I am not really sure about this.
- Same as before: If I could know enough to expand into n-dimensions and come up with simple variations on the topics discussed in class, I will be satisfied. The wave equation would be a good goal for a first attempt at FEMs.
- If we could understand the basic idea then it will not be very difficult to expansion the code to solve different types of PDE. I expect the we may be able to use our fundamental package to solve linear , nonlinear PDE and even system of PDEs .
- most kind of equation in the future
- I hope to be able to solve most second-order PDEs for fairly complex boundary conditions.

Pre-assessment

What is your expectation from this course for improving your general capability in coding (not necessarily just for finite element methods – such as the number of rows or the complexity of the code you will be able to handle by yourself)?

- I expect to learn the basics of programming from a mathematical standpoint: plotting a function in different forms to an extent, solving systems of equations, and understanding how to set up an iterated process.
- I expect very little from this course since I've already taken numerous programming/software engineering courses.
- MATLAB is not a complex language, I don't have high expectations here.

Pre-assessment

- I think it would help a lot. Learning and understanding this code package is really a great help in future research
- I see this course as a different viewpoint to coding I have done previously. It will make me a stronger coder, but not necessarily with regards to code complexity.
- I hope we can improve our ability to write coding and programming and advanced our skills to dealing with different kind of PDEs
- cod eeasily and efficiently
- I expect to get a better understanding of the general workings of MATLAB, including think like file input/output and manipulating matrices.

Pre-assessment

What other expectations or comments on this course do you have before you take it?

- I expect to have trouble understanding the material the first few weeks, and then to catch up as I come to understand the terminology.
- Having taken an engineering finite element course, I felt like I really didn't learn very much about the finite element method itself. I could solve problems similar to the in-class examples and make limited use of commercial code, but wouldn't have any idea how to code finite element software on my own or even adapt to more complex problems. I'm excited that this course will try to build an understanding of the basic method and then give input on how to generalize it and turn it into code.

Other assessments

- Mid-assessment: March 16
- Post-assessment: May 8

Current status

- Both lectures and guided coding are going well;

Current status

- Both lectures and guided coding are going well;
- Homework 1: 13 out of 18 students successfully finish the code package;

Current status

- Both lectures and guided coding are going well;
- Homework 1: 13 out of 18 students successfully finish the code package;
- Homework 2: 15 out of 18 students successfully finish the code package;

Current status

- Both lectures and guided coding are going well;
- Homework 1: 13 out of 18 students successfully finish the code package;
- Homework 2: 15 out of 18 students successfully finish the code package;
- Midterm project: 17 out of 18 students successfully finish the code package;

Outline

Conclusions

- Provide a bridge between the traditional mathematical finite element courses and the traditional engineering finite element courses

Conclusions

- Provide a bridge between the traditional mathematical finite element courses and the traditional engineering finite element courses
- Need to redesign the whole course: lectures, guided coding, homework, projects, and independent study.

Conclusions

- Provide a bridge between the traditional mathematical finite element courses and the traditional engineering finite element courses
- Need to redesign the whole course: lectures, guided coding, homework, projects, and independent study.
- Train the students for (a) a thorough understanding of finite element; (b) the capability to self-implement finite element methods to independently form a new finite element package under a general framework; (c) the capability to correctly and quickly understand and modify open-source packages for new problems.

Conclusions

- Provide a bridge between the traditional mathematical finite element courses and the traditional engineering finite element courses
- Need to redesign the whole course: lectures, guided coding, homework, projects, and independent study.
- Train the students for (a) a thorough understanding of finite element; (b) the capability to self-implement finite element methods to independently form a new finite element package under a general framework; (c) the capability to correctly and quickly understand and modify open-source packages for new problems.
- Good for students with different types of students with or without experience in coding and finite element methods.