Introducing the Galerkin Method of Weighted Residuals into an Undergraduate Elective Course in Finite Element Methods

Dr. Aneet Dharmavaram Narendranath, Michigan Technological University

Dr. Aneet Dharmavaram Narendranath is currently a Lecturer at Michigan Technological University (Michigan Tech). He received a PhD in Mechanical Engineering-Engineering mechanics in 2013. Subsequently, he worked as a visiting assistant professor at Michigan Tech from 2013-2014 and then as an Engineer at the French Nuclear Commission (CEA) in France until 2015. His research interests are mathematical modeling of fluid physics. His pedagogical interests are development of mathematically oriented courses in mechanical engineering.
Introducing the Galerkin method of weighted residuals into an undergraduate elective course in finite element methods

Abstract

Modern day finite element methods (FEM) are closely attached to the advent of mathematical and matrix algebra methods in the design of aeronautical structures. Primarily, FEM is an approximation technique for partial differential equations. The power of FEM is realized when the fundamental field problems governing the engineering design are “encompassed” in irregular shapes. In other words, FEM is particularly useful in resolving the effect of static or dynamic loads (structural or thermal) on complex shapes. In this paper, regular shapes are: square/rectangular geometrics, circular cross sections.

Modern day finite element method (post 1940s-50s) as taught in undergraduate level (senior level) electives shows bifurcation from classical methods (pre 1900s) in at least its abstraction from rigorous mathematical concepts through the use of powerful software tools. However, it is beneficial for students of FEM to be made aware of the connection between classical methods (differential equations) and computer tool based analysis.

The overall objective is to introduce the Galerkin method of weighted residuals for linear ordinary differential equations and to extend that idea to linear, steady state problems in structural mechanics and thermal transport. Exposure to the Galerkin method allows students to connect differential equation based mathematical models to plane-problems in elasticity, lubrication theory problems in fluid dynamics and steady state thermal transport problems. Students are made aware of the concept of “global” vs “local” shape functions, “element order”, “convergence” and “error”.

The Poisson’s equation $u''(x)=f$ is primarily utilized to build the students’ confidence in solving differential equations and applying the Galerkin method. This allows students to forge a connection between differential equations and simple linear (yet powerful) mathematical models. An incremental approach is taken by making the Poisson’s equation heterogenous from homogenous (i.e, $f\neq0$ or $f=f(x)$ from $f=0$). Students find appropriate polynomial functions for use in the Galerkin method of weighted residual for the Poisson’s equation. The choice and order of polynomial functions and its relation to modifying or refining a shape function in software is realized.

Finally, MATLAB and its partial differential equation toolbox, pdetool, is used to connect the Galerkin Method to classical engineering problems. How boundary conditions could have an effect of reducing a 2-D problem to a 1-D problem was explored. This exercise allowed students to be conscientious of boundary conditions and the variety and applicability thereof, as evidenced through examination and homework assignment results.
Homework assignments, examinations, end of semester design problem/project and student exit surveys are used as metrics to check efficacy of pedagogy. This course on finite element methods targets ABET criteria a,b,e,g,i,k.

**Paper Outline**

This paper describes (i) analytical mathematical techniques, viz., solution of differential equations by the method of variables separable and Galerkin’s method of weighted residuals and (ii) computational tools, viz, MATLAB and its partial differential equations toolbox (pdetool) for an undergraduate elective course in finite element methods.

In this paper, an introduction, literature review and brief philosophy of this study and the class demographics are first described. A skill assessment exam is conducted to quantitatively uncover the lack of understanding and application of boundary conditions as prevalent in most students. A description of the equation of choice and the rationale for this follows along with a list of examples that students have at their disposal to correct their deficiency in understanding the significance of and application of boundary conditions. This short yet powerful list of examples also sheds light on the fundamentally important problems in solid mechanics and fluid/thermal transport. A description of one of the final skill assessment examination problems follows along with a discussion. Sample student results are then depicted. Only one correct solution is chosen for this description whilst 4 wholly or partially incorrect solutions are also depicted. Finally, as supplementary material, examples using MATLAB pdetool is provided for readers to adopt or adapt for their respective courses and code snippets for post processing are provided to visualize, parse, interpolate and interpret data.

**Introduction and literature search**

The author performed a literature search (table 1,2) for the state of the art in the utility of the Galerkin’s method of weighted residuals in a predominantly undergraduate engineering classroom. The literature search included some prominent textbooks in the Finite Element Method (FEM), ASEE publications that appear through the use of search parameters “FEM/finite element”, “Galerkin” or “method of weighted residual” and the relevant “International journal of mathematical education in science and technology” and the American physical society’s “the Physics teacher”. It was revealed from this literature search, that the scope defined in this study was relevant and different from what is in literature.

1. Previous authors have applied variational calculus concepts for the use of the Galerkin’s method. However, since variational calculus itself is outside the scope of the current undergraduate course in FEM, the author has utilized the Galerkin’s method of weighted residuals as demonstrated by Duncan. This method does not require an understanding of weak and strong formulations and is perhaps one of the first treatises on the Galerkin method in English.

2. Previous authors have made no explicit connection between the direct stiffness method and
a method of weighted residuals. Both lead to condensation of a problem into the
\( \{ F \} = [K]\{x\} \) form. This latter linear matrix algebra form is what modern day FEM hinges on.

3. Previous authors, when the Galerkin method is used, have made no explicit connection between increasing the order of a shape function polynomial and the refinement of a mesh.

4. The use of MATLAB’s pdetool for an FEM class has not been found.

5. At least one previous author mentions the correct balance between theory and software practice for undergraduate courses in FEM as being an important criteria. The current paper finds a common thread in the tapestry of FEM and explores it using theory (Galerkin MWR) and software (MATLAB pdetool).

In summary in light of the literature search, the current paper explores a balance between theory and software practice in FEM through the application of the Galerkin method of weighted residuals. Supplementary, but important addition are the recognition of a class of differential equations for a wide variety of fluid-thermal transport and structural mechanics problems and the application of boundary conditions and validation of numerical results.

<table>
<thead>
<tr>
<th>Textbook</th>
<th>Method of analysis</th>
<th>Deficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reddy(^2)</td>
<td>Use of the weighted residuals through variational calculus and strong (\rightarrow) weak formulation.</td>
<td>Students do not have background of variational calculus. No connection provided between direct stiffness method and the MWR. Higher order shape function utilization in the MWR and its connection to higher order 2D elements not made in this book.</td>
</tr>
<tr>
<td>Rao(^3)</td>
<td>Only linear polynomial piecewise shape functions are used.</td>
<td>No connection provided between direct stiffness method and the MWR. Higher order shape function utilization in the MWR and its connection to higher order 2D elements not made in this book.</td>
</tr>
<tr>
<td>Logan(^4) (used as primary textbook in current course)</td>
<td>Sparse explanation of MWR and its applicability</td>
<td>Higher order shape function utilization in the MWR and its connection to higher order 2D elements not made in this book. Applications limited to simple bar elements.</td>
</tr>
</tbody>
</table>

Table 1: Galerkin’s method of weighted residuals (MWR) in some prominent textbooks on the finite element method and their deficiency

As part of an elective course in finite element methods, the author instructs the use of the Galerkin method of weighted residuals (Galerkin MWR) to solve fundamentally important problems in fluid-thermal transport and solid mechanics. The usual topics of direct stiffness method and
<table>
<thead>
<tr>
<th>Publication</th>
<th>Method of analysis</th>
<th>Deficiency or opportunity for improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>International Journal of Mathematical Education in Science and Technolog The Physics teacher</td>
<td>No mention of use of Galerkin MWR</td>
<td>No mention of use of Galerkin MWR</td>
</tr>
<tr>
<td>Monterrubio⁵</td>
<td>Use of the Rayleigh-Ritz variational method for structural (beam) problem</td>
<td>Author does not not use the MWR (although the Ritz method can be equivalent or produce equivalent results).</td>
</tr>
<tr>
<td>Hamann⁶</td>
<td>Use of MATLAB for hyperbolic partial differential equations</td>
<td>Course for graduate students in partial differential equations. The weighted residual method as a solution method is only expressed as part of the introduction.</td>
</tr>
<tr>
<td>Walker⁷</td>
<td>Galerkin weak formulation</td>
<td>Walker⁷ Does not explore the parallels between the direct stiffness method and the Galerkin method and the use of MATLAB’s pdetool and its efficacy in the realization of boundary condition formulation.</td>
</tr>
<tr>
<td>Echempati⁸</td>
<td>Assesment of an FEM course and focus on key characteristics of an FEM course for undergraduates</td>
<td>Mentions that FEM textbooks should strike a balance between theory and software practice. The prominent textbooks perused for this study (table</td>
</tr>
</tbody>
</table>
software skills are also instructed. The idea was to expose students to the powerful technique of Galerkin MWR along with traditional tools of approximation.

This study of the introduction of the Galerkin’s method of weighted residuals (Galerkin’s MWR) into a predominantly undergraduate course in finite element methods (FE) is multi-fold:

1. A recognition of an approximation method put forth by Galerkin to:
   
   (a) An appreciation of various trajectories that have led to the modern day FEM formulation of problems into an $\{F\} = [K]\{X\}$ form, that is relatively easily solved using computers. In effect, the equivalence of the “direct stiffness method” to that of the “method of weighted residuals by Galerkin” allows for students to see FEM as an approximation technique for (mechanical) engineering “field problems”.

   ![Diagram of Direct Stiffness Method and Method of Weighted Residuals](image)

   Figure 1: Equivalence of the direct stiffness method and the Galerkin method of weighted residuals (MWR). Both trajectories of practice lead to the matrix structural formulation of $\{F\} = [K]\{X\}$. This comparison, although part of this course, is not included in this paper to keep it short.

   (b) A recognition of various boundary conditions and their relevance and similarity in solving plane problems in elastic theory (mechanics of materials), low Reynold’s number flow problems in fluid dynamics, steady state thermal transport problems and vibration problems. This study utilizes a small set of boundary conditions and mathematical models that they would have to source from for simple problems in solid mechanics (static and dynamic-vibrations) and fluid thermal transport (one-dimensional steady state lubrication approximation and one-dimensional steady state thermal transport). Along the way, students are also exposed to the ideas of using fluid mechanical continuity conditions as boundary conditions and the use of boundary condition to model a one-dimensional problem on a two-dimensional grid (for one-dimensional fluid mechanics based on lubrication approximation).

   (c) Validation of FEM results for fundamental problems, with analytical results and an appreciation of the notion of shape functions as approximating polynomials and their effect on simulation results.
2. Using the knowledge gained on this approximation method in the application of proper boundary conditions and utility of MATLAB’s *pdetool* (part of the “Partial Differential equations Toolbox”) to solve plane problems in elasticity, fluid-thermal transport using modern day FEM (use of 2-D elements, exploring element quality, exploring convergence and grid dependency).

3. Disseminating information and diverse education comprising of conceptual understanding and hands-on skills to students to ensure conformation with various items in the department’s strategic plan.

Students must understand the relationship between various mathematics based courses and core mechanical engineering courses. This allows for a greater appreciation of computer software\[13\], its ability (and lack of) and when one may need to or not- use it. Students also map the steps involved in Galerkin’s MWR with the general steps involved in the use of software (see table\[3\]).

<table>
<thead>
<tr>
<th>Analytical Step</th>
<th>Equivalent step in software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formulate problem statement</td>
<td>Create geometry and apply BCs</td>
</tr>
<tr>
<td>(differential equation with BCs)</td>
<td>(including loads, which are a Neumann BC)</td>
</tr>
<tr>
<td>Choose an approximating global shape function</td>
<td>Discretize the geometry (there are rules/guidelines involved)</td>
</tr>
<tr>
<td>(there are rules involved)</td>
<td>“Solve the problem”</td>
</tr>
<tr>
<td>Reconcile coefficients in global shape function</td>
<td>“Post processing and plotting”</td>
</tr>
<tr>
<td>Plot the field variable solved for in the problem</td>
<td>Validate numerical solution with analytical solution or</td>
</tr>
<tr>
<td>statement with the coefficients included</td>
<td>experimental results</td>
</tr>
</tbody>
</table>

Table 3: Mapping of Galerkins MWR to software use.

This study does involve students solving fundamentally important engineering problems using MATLAB’s *pdetool* and validating these results using the Galerkin MWR. Moreover, the reason that the mathematical theory of Galerkin’s MWR is used is so that those students who wish to pursue advanced study in finite element method are provided some initial direction. In fact, many students from this course do register for a graduate level course in finite element methods.

**Class demographics**

The students who enroll for this course in FEM are either 3rd or 4th year mechanical engineering undergraduate students or 1st year graduate students in mechanical engineering, electrical engineering or civil engineering. This course includes rigor pertinent to both levels of students viz., senior level undergraduate students and first year graduate students.
This undergraduate level elective in finite element methods (FEM) is predominantly constituted of by undergraduate students (approximately 70% of the course enrollment). There are graduate level students in this course as well (approximately 30% of the course enrollment) as it informs them of skills that they can apply to many graduate level courses (such as computational fluid dynamics, numerical methods for differential equations, advanced finite element methods) which have a primary or secondary focus on numerical methods and approximations. This course is offered throughout the academic year in Fall (September-December), Spring (January-May) and track-B of Summer (June-Aug). The idea is to expose students to multiple trajectories of FEM and the equivalence of these trajectories.

**Assessment of skills**

The skills of students entering this course are assessed through an initial “examination” that tests two aspects. The total number of students in this analysis was 34. Although this examination is not graded, the students receive feedback on any errors they may have committed. This examination also allows for the instructor to pay attention to fundamental details that are lacking in the students’ knowledge.

1. **Analytical skills:** Application of boundary conditions for a second order linear ordinary differential equation for the following problem:

   Consider a uniform rod subjected to a uniform axial load $q_0$. This rod is fixed at $x = 0$ and is free at $x = L$. It can be readily shown that the deflection, $u(x)$ of this bar element is described by the governing second order linear ordinary differential equation:

   $$\frac{d^2 u}{dx^2} + q_0 = 0$$

   (1)

   Find an exact solution for the deflection through the application of correct boundary conditions. (Students are required to realize and apply the following boundary conditions: $u(x = 0) = 0$ and $\frac{du}{dx}|_{x=L} = 0$ to resolve the two constants of integration when the method of variable separable is used.)

2. **Software skills:** The use of software (ANSYS) to solve a simple beam deflection problem by following a set of steps.

   Majority of (24) the students in this course were unsuccessful in the use and proper application of boundary conditions analytically. However, a majority of the students (30) were able to use ANSYS to solve the statically determinate loaded beam problem. Statistical correlations between student success in the analytical skills portion with the software skills portion was not developed at this stage. Based on the students’ unsuccessful attempt at reconciling boundary conditions mathematically, the instructor used a sequential presentation of facts to ensure students “connected the dots” and recognized the synergy between basic mathematical methods and software based methods, for fundamental engineering problems.
Galerkins MWR: Notes and examples

Notes

The instructor has collected multiple classical papers on the Galerkins MWR and distilled from these notes steps on how to apply this MWR for a particular field problem (differential equation) of importance. The differential equation used is the second order Poisson’s equation (one-dimensional) as shown in equation \( \frac{d^2 u}{dx^2} = f \) that has broad utility in mechanical engineering and applied physics. The students are also conversant with this equation having seen it on multiple occasions in introductory courses on ordinary differential equations (ODEs). All students enrolled in this course have had ODEs.

\[
\frac{d^2 u}{dx^2} = f
\]  

Examples

The Poisson’s equation has been used to develop multiple examples in fluid mechanics, thermal transport and solid mechanics (with appropriate boundary conditions). The student applies the Galerkin MWR on these examples to:

1. Develop approximate solutions. (skill exercised: exposure to idea of approximation)
2. Compare the approximate solution with exact solution. (skill exercised: checking for convergence)
3. Modifies the weight function in the Galerkin method to iteratively develop a better approximation if necessary. (skill exercised: changing the order of the shape function/element/approximating polynomial for higher accuracy.)
4. Solution of problem using MATLAB’s native FEM solver, pdetool. The MATLAB pdetool allows for the solution of two-dimensional problems in elasticity, fluid-thermal transport and electromagnetism. Appropriate boundary conditions would need to be applied to have it solve one-dimensional problems as modelled by the Poisson’s equation. Post-processing of results and checking with Galerkin MWR. (skill exercised: Comparison of numerical solution with analytical/semi-analytical results).

In case of fluid mechanics, the Poisson’s equation (equation \( \frac{d^2 u}{dx^2} = f \)), describes steady state, one-dimensional pressure driven \((f = \frac{dp}{dx})\) Poiseuille flow or shear driven Couette flow \((f = 0)\). In case of heat transfer, equation \( \frac{d^2 u}{dx^2} \) describes steady state, one-dimensional conduction with \(f = g(x)\) or without a heat source \(f = 0\). In solid mechanics and elastic theory, the Poisson equation describes a one-dimensional link under a loading condition described by \(f\). In two dimensions, this equation may also be used to represent torsion of shafts (solid mechanics) or potential flow describing flow over a cylinder or sphere (fluid mechanics, important for studying aerodynamics, boundary layers etc.). However, for this study, only one-dimensional form of the Poisson’s equation is solved analytically.
The boundary conditions (see table 4) mainly used for this problem are Dirichlet or Neumann. This terminology (Dirichlet and Neumann) is used regularly in this course so that student are eventually also able to make appropriate recognitions of boundary conditions when using MATLAB and its native FEM solver, pdetool. MATLAB’s pdetool uses this terminology and it is necessary for students to be conversant with it.

<table>
<thead>
<tr>
<th>Boundary condition</th>
<th>Formulation</th>
<th>Field</th>
<th>Physical significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirichlet</td>
<td>$u</td>
<td>_{x=\xi} = T_0$</td>
<td>Thermal transport</td>
</tr>
<tr>
<td>Dirichlet</td>
<td>$u</td>
<td>_{x=\xi} = u_0$</td>
<td>Fluid mechanics</td>
</tr>
<tr>
<td>Dirichlet</td>
<td>$u</td>
<td>_{x=\xi} = x_0$</td>
<td>Solid mechanics</td>
</tr>
<tr>
<td>Neumann</td>
<td>$du/dx</td>
<td>_{x=\xi} = q_0$</td>
<td>Thermal transport</td>
</tr>
<tr>
<td>Neumann</td>
<td>$du/dx + dv/dy</td>
<td>_{x=\xi} = q_0$</td>
<td>Fluid mechanics</td>
</tr>
<tr>
<td>Neumann</td>
<td>$du/dx</td>
<td>_{x=\xi} = f_0$</td>
<td>Solid mechanics</td>
</tr>
</tbody>
</table>

Table 4: Boundary conditions for the Poisson’s equation

MATLAB pdetool examples

The following examples have been developed using MATLAB and its pdetool function. They are utilized to describe problem set-up and solution in MATLAB’s pdetool. These examples are also performed by students in a commercial FEA software.

1. **Plane problems in elasticity: Simply supported beams, cantilever beams and axial members.** These are solved in two-dimensional space for a one-dimensional field variable (displacement). Checks may be performed using exact solutions, Galerkin method may be used also. Student needs to choose the appropriate differential equation and choose appropriate boundary conditions (Dirichlet, Neumann with proper values for deflection or load).

2. **Pressure drive Poiseuille flow problem.** This is again a one-dimensional flow problem solved in two-dimensional space. Checks are performed using exact solutions, Galerkin method is also used. Student needs to choose the appropriate differential equation and choose appropriate boundary conditions (Dirichlet for top and bottom walls, Neumann/continuity equation for inlet and outlet). This specific problem is interesting as none of the students realized (checked through a show-of-hands of 12/12 students (12 students out of 12 present on the day) in Summer 2016 and 22/22 students (22 students out of 24 present on the day) in Fall 2016) that the Continuity condition needs to be used as a “boundary condition” defining the continuity of flow at the inlet and outlet of this flow problem. The solution obtained using MATLAB’s pdetool is shown in figure 3.

3. **Thermal transport: one-dimensional heat conduction with heat source.** This is a one-dimensional heat conduction problem that is solved in two-dimensional space. The Galerkin MWR is used to solve this non-homogenous problem with Dirichlet boundary
conditions. Multiple iterations are required to achieve convergence with exact solution. Neumann conditions of zero heat flux are require to solve this problem in two-dimensional space. This specific problem is also interesting as none of the students realized (checked through a survey of 12/12 students (12 students out of 12 present on the day) in Summer 2016 and 22/22 students (22 students out of 24 present on the day) in Fall 2016) that the Neumann/heat flux condition needs to be used as a “boundary condition” to solve this one-dimensional problem in two-dimensional space. Effectively, heat flux in one direction has to be set to zero to ensure one-dimensional isotherms. The simulation results from MATLAB’s pdetool are shown in figures 18 and 19.

4. Supplementary simulations. MATLAB pdetool example programs have also been developed to study inviscid flow over cylinders (potential flow: uses Poisson’s equation) and stress concentration in a “plate with a hole in tensile field”. These are optional materials available for students to practice their MATLAB simulation, data processing and analytical skills on. The important result of this supplementary materials is to show the similarity of pressure gradients and stress concentrations in these problems. They are both amenable to potential theory. They are both solved using the same or similar differential equations. These may be utilized in future semesters.

Note that the student also develops experience and expertise with a more standard commercial FEM package, viz. ANSYS for solutions of a field equation on complex geometries. MATLAB’s pdetool also does allow for complex geometries, through either the creation of geometry through manually using primitive shapes (lines, points, curves etc.) or through the importing of .stl files.
Figure 3: Poiseuille flow (pressure driven laminar flow) problem solved using MATLAB’s pdetool. The parabolic velocity profile obtained using this FEM tool is compared with the solution obtained using Galerkin’s method of weighted residuals. Exceptionally good match between results (nearly 100%) is achieved even with an “out of the box” coarse mesh.

Final assessment of skills

The final assessment of skills learned was based on the “Thermal transport: one-dimensional heat conduction with heat source” discussed in the section on “MATLAB pdetool examples”. The problem statement and the differential equation is provided below. The boundary conditions must be designed by the student for both the exact/Galerkin MWR and the MATLAB pdetool driven solution.

Consider the unsteady 1-D heat equation for temperature \( u \) [in units of Kelvin].

\[
\rho C \frac{du}{dt} - k \frac{d^2u}{dx^2} = Q + h (u_{ext} - u) \tag{3}
\]

For the following example values: \( \rho = 1.0 \) [kg/cubic m], \( C = 1.0 \) [J/kg-K] and \( k = 1.0 \) [W/m-K], with a heat source term \( Q = x \) [W/cubic m] and no convection heat transfer, with boundary conditions \( u(0) = 0 \) and \( u(1) = 1 \) on the domain \( (0, 1) \):

- For steady state conditions, solve and specify an exact solution to this mathematical model describing the temperature profile.

- Devise a polynomial based global shape function of the, use the Galerkin method of weighted residuals to find the approximate solution \( u_N \). Refine your shape function so that the percentage error between the exact solution and the MWR approximation is less than 5%.

- Solve this problem using MATLAB and its pdetool capability and upload your MATLAB source file to canvas. Use the following mesh input parameters described in table 5.
Figure 4: Classical elasticity problem of “plate with hole in a tensile field” (horizontal stress field is plotted) on the left and the “inviscid flow over a cylinder” fluid mechanics problem (pressure distribution is plotted) on the right. These are described to students through MATLAB’s pdetool. Validation of the stress field around the hole (left) and the pressure field around the cylinder (right) may be performed using either a stress function approach or the Bernoulli’s equation, respectively. This may be used in future semesters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh Maximum edge size</td>
<td>0.05</td>
</tr>
<tr>
<td>Mesh growth rate</td>
<td>1.3</td>
</tr>
<tr>
<td>Mesher version</td>
<td>Default value</td>
</tr>
<tr>
<td>Jiggle mesh</td>
<td>Uncheck (deselect)</td>
</tr>
</tbody>
</table>

Table 5: Mesh initialization parameters that students must use

**Discussion**

Initially a large number of students were unable to determine the boundary conditions needed for an “axial member under load” problem but were able to effectively use ANSYS to solve similar problems. An explicit cognizance of boundary conditions was missing. Finally, the thermal transport problem (with heat source) needed students to recognize and apply, independently, the following:

1. **Choice of appropriate mathematical model (differential equation) and associated boundary conditions.** This shows a cognizance of the mathematics behind the problem.

2. **Proper application of the Galerkin MWR to solve for temperature field.** This shows the development of skills that allow for the choice of proper shape functions for approximations.

3. **Validation:** Exact solution of the temperature field and comparison with the approximation from the Galerkin MWR. This allows for a cognizance and appreciation of exact solution (where available).
4. **Refinement of approximation** by choosing a higher order shape function and re-solving for temperature field until convergence to within a specified percentage error. This shows that students need to validate their numerical/FEM results.

5. **Solution of this problem using MATLAB pdetool** whilst being cognizant that the problem statement is one-dimensional but MATLAB’s *pdetool* allows only two-dimensional problems. Appropriate application of insulated (Neumann) boundary condition to convert two-dimensional *pdetool* specification to mimic one-dimensional thermal flow. This shows a refined level of expertise where boundary conditions are used in an advantageous fashion. It also shows that students have an understanding of the underlying physics. They are looking for one-dimensional isotherms.

6. **Postprocessing of FEM results available from pdetool and comparison with exact solution and Galerkin MWR approximation.** Presentation of results and reasoning skills of students are exercised.

**Learning and Teaching styles**

Learning and teaching styles that this study used, according to Felder et al.[4]

1. Students are exposed to the method of weighted residuals through **sequential steps**. An equivalence of the direct stiffness method of weighted residuals is demonstrated through **visual presentations**.

2. **Inductive reasoning** is used. Facts about the MWR and observations that related to the equivalence of the MWR to the direct stiffness method are provided and underlying principles are inferred.

3. Initial acquisition of knowledge by the students is **passive**. They apply the knowledge acquired about the MWR (**active learning**) to solving fundamentally important problems in elastic theory and transport theory.

**Sample student results**

**Exact solution compared with Galerkin method**

Students compared the exact solution with the Galerkin MWR. The idea was to reinforce the effect of the approximating polynomial (global shape function). Students realized that changing the order of the polynomial allowed for a better degree of accuracy of the Galerkin MWR with respect to the exact solution.
Results from MATLAB pdetool

MATLAB’s pdetool was used to simulate the same problem (equation 3) and the results show qualitative equivalence with the Galerkin method and the exact solution. Students have qualitatively (as shown in figure 8) and quantitatively (figure 9) compared the temperature profile obtained using MATLAB’s pdetool with the analytical methods (Exact solution obtained using method of variables separable and Galerkin MWR). To quantitatively compare results obtained from MATLAB’s pdetool with the exact solution, one must use the MATLAB internal function pdeInterpolant. The MATLAB code for this exam problem along with post-processing and pdeInterpolant code snippets are included at the end of this text for use by other instructors and students.
Figure 6: Steady state temperature at equidistant spatial locations.

<table>
<thead>
<tr>
<th>Spatial loc.</th>
<th>Temperature (exact)</th>
<th>Temperature (2nd order P)</th>
<th>Temperature (3rd order P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.0820625</td>
<td>0.0816667</td>
<td>0.0820625</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1615</td>
<td>0.16</td>
<td>0.1615</td>
</tr>
<tr>
<td>0.15</td>
<td>0.238188</td>
<td>0.235</td>
<td>0.238188</td>
</tr>
<tr>
<td>0.2</td>
<td>0.312</td>
<td>0.306667</td>
<td>0.312</td>
</tr>
<tr>
<td>0.25</td>
<td>0.382813</td>
<td>0.375</td>
<td>0.382813</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4505</td>
<td>0.446915</td>
<td>0.4505</td>
</tr>
<tr>
<td>0.35</td>
<td>0.514938</td>
<td>0.507169</td>
<td>0.514938</td>
</tr>
<tr>
<td>0.4</td>
<td>0.576</td>
<td>0.5615</td>
<td>0.576</td>
</tr>
<tr>
<td>0.45</td>
<td>0.633563</td>
<td>0.615997</td>
<td>0.633563</td>
</tr>
<tr>
<td>0.5</td>
<td>0.6875</td>
<td>0.672837</td>
<td>0.6875</td>
</tr>
<tr>
<td>0.55</td>
<td>0.737687</td>
<td>0.715321</td>
<td>0.737687</td>
</tr>
<tr>
<td>0.6</td>
<td>0.784</td>
<td>0.762433</td>
<td>0.784</td>
</tr>
<tr>
<td>0.65</td>
<td>0.826313</td>
<td>0.801667</td>
<td>0.826313</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8645</td>
<td>0.851082</td>
<td>0.8645</td>
</tr>
<tr>
<td>0.75</td>
<td>0.898438</td>
<td>0.876028</td>
<td>0.898438</td>
</tr>
<tr>
<td>0.8</td>
<td>0.928</td>
<td>0.907667</td>
<td>0.928</td>
</tr>
<tr>
<td>0.85</td>
<td>0.953862</td>
<td>0.935666</td>
<td>0.953862</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9735</td>
<td>0.960636</td>
<td>0.9735</td>
</tr>
<tr>
<td>0.95</td>
<td>0.998187</td>
<td>0.984667</td>
<td>0.998187</td>
</tr>
<tr>
<td>1.</td>
<td>1.</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 7: Percentage error using the Galerkin MWR for the steady state heat equation with heat source. Students show that a third order polynomial approximation (global shape function) produces the best results as validated through comparison with exact solution.
Figure 8: Temperature profile from MATLAB pdetool (left) compared qualitatively with analytical methods (right, Galerkin and separation-of-variables-exact).

Figure 9: Temperature profile from MATLAB pdetool compared quantitatively with analytical methods.
MATLAB *pdetool* selection screens

This section briefly depicts the various selection screens available in MATLAB’s pdetool. It is quite close in operation to the general preprocessing, solution and postprocessing steps and screens available in most FEM software packages.

![Options available in pdetool](image_url)

Figure 10: Options available in *pdetool*. It is customary to start with *Draw* and then move on to *PDE specification*, *Boundary condition specification*, *Mesh properties*, *Solution*, *Exporting solution for postprocessing*.

![Problem domain constructed using Draw options in pdetool](image_url)

Figure 11: Problem domain constructed using *Draw* options in *pdetool*
Figure 12: Mathematical model (PDE) specification for the problem in equation 3.

Figure 13: Dirichlet boundary condition (constant temperature) specification for equation 3.
Neumann insulated conditions when specified for the upper and lower boundaries allow for this two-dimensional problem to be solved as a one-dimensional problem. The results show one-dimensional isotherms as are apparent from the plots on temperature contours (fig 18).

Figure 14: Neumann boundary condition (insulated temperature) specification for equation 3.

Figure 15: Default mesh. Students also realized through internal options that by default, the mesh constructed by pdetool is a first order mesh (constant strain triangles).
Figure 16: Higher mesh density. Students performed mesh refinement as available in pdetool. It is a simple button click that translates to using more elements while ensuring that a generally high mesh quality is retained.

Figure 17: Mesh quality plot for the case with “high mesh density” as in figure 16.
Figure 18: Temperature contours. Default mesh. The contour plot provides qualitative validation of the boundary conditions being satisfied. The vectors/arrows depict the direction of heat flux.

Figure 19: Temperature contours. Higher density mesh. The isotherms are straighter with a high density mesh.
Conclusion

This paper describes how the Galerkin method of weighted residuals (Galerkin MWR) was introduced into an undergraduate elective course in finite element methods (FEM). This elective is open to both senior level undergraduate students and beginning graduate students. Beginning graduate students enroll in this course as a stepping stone to graduate level finite element method courses. This course instructs students on the use of commercial FEM software (ANSYS) though a lab section (2 hours per week). This course also has lectures associated with it (2 hours per week). The focus of the lectures is analytical methods and the theory behind FEM.

As part of the lectures, the instructor observed that the majority of (24) the students in the beginning of this course were unsuccessful in the proper application of boundary conditions and solution of a differential equation describing linear bar deflection. However, a majority of the students (30) were able to use ANSYS to solve the statically determinate loaded beam problem.

Through the utilization of classical publications on the Galerkin MWR, the instructor included this in the lectures. The idea was to have students come to a realization of the parallels between this powerful analytical approximation technique (Galerkin MWR) and modern day FEM. Students used both the Galerkin MWR and modern day FEM tools (MATLAB pdetool) to solve a nonhomogenous, one dimensional, steady state thermal transport problem. As part of this exercise, the following were revealed:

- The utility of boundary conditions to model a 1-D problem on a 2-D domain (because MATLAB pdetool allows only 2-D domains).
- The exact solution of the differential equation that governs the physical problem of thermal transport with the appropriate use of boundary conditions.
- The solution of the governing differential equation through the Galerkin MWR and comparison with the exact solution for different approximation polynomial orders.
- Qualitative and quantitative comparison of the solution achieved for this thermal transport problem through MATLAB pdetool with the exact solution and the Galerkin MWR solution.

It was revealed through this exercise that 31 out of 34 students were able to accomplish this problem successfully with no direction from the instructor. The remaining 3 students were also able to complete this problem but required regular meetings and direction from the instructor. Their initial unsuccessful attempts at this problem have been captured in the appendix.

Acknowledgement

The author acknowledges the department of mechanical engineering-engineering mechanics at Michigan technological university for support provided to the instructor to teach this course on multiple occasions. This allowed for a steady improvement in examples being used in class to fortify concepts for students. The author thanks ASEE reviewers and the chair for comments,
suggestions and recommendations provided during various stages of preparing this paper. This input allowed for an improvement in the structure of this paper.

References


APPENDIX

MATLAB code for exam problem

All these example problems may be downloaded at [https://github.com/dnaneet/ASEE17](https://github.com/dnaneet/ASEE17).
% This script is written and read by pdetool and should NOT be edited.
% There are two recommended alternatives:
% 1) Export the required variables from pdetool and create a MATLAB script
%  to perform operations on these.
% 2) Define the problem completely using a MATLAB script. See
% http://www.mathworks.com/help/pde/examples/index.html for examples
% of this approach.

function pde_model
[pde_fig,ax]=pdeinit;
pdetool('appl_cb',9);
set(ax,'DataAspectRatio',[1 0.91271033653846145 1]);
set(ax,'PlotBoxAspectRatio',[1.6434567901234569 1.0956378600823047 1]);
set(ax,'XLimMode','auto');
set(ax,'YLimMode','auto');
set(ax,'XTickMode','auto');
set(ax,'YTickMode','auto');

% Geometry description:
pderect([0 1 0.5 0],'R1');
set(findobj(get(pde_fig,'Children'),'Tag','PDEEval'),'String','R1')

% Boundary conditions:
pdetool('changemode',0)
pdesetbd(4,...
'dir',...  
'1',... 
'1','... 
'0')
pdesetbd(3,...
'neu',...
'1',...
'0',...
'0')
pdesetbd(2,...
'dir',...  
'1',...
'1','... 
'1')
pdesetbd(1,...
'neu',...
'1',...
'0',...
'0')

% Mesh generation:
setappdata(pde_fig,'Hgrad',1.3);
setappdata(pde_fig,'refinemethod','regular');
setappdata(pde_fig,'jiggle',char('on','mean',''));
setappdata(pde_fig,'MesherVersion','preR2013a');
pdetool('initmesh')
pdetool('refine')
pdetool('refine')
% PDE coefficients:
pdeseteq(1,...
'1.0',...
'0',...
'(x+1)+(0.0)+(0.0)',...
'(1.0)+(1.0)',...
'0:10',...
'0.0',...
'0.0',...
'[0 100]')
setappdata(pde_fig,'currparam',... 
['1.0';...
'1.0';...
'1.0';...
'x+1';...
'0 ';...
'0.0'])

% Solve parameters:
setappdata(pde_fig,'solveparam',... 
char('0','4032','10','pdeadworst',... 
'0.5','longest','0','1E-4','','fixed','Inf'))

% Plotflags and user data strings:
setappdata(pde_fig,'plotflags',[1 1 1 2 1 7 1 0 0 0 1 1 0 1 0 1]);
setappdata(pde_fig,'colstring','');
setappdata(pde_fig,'arrowstring','');
setappdata(pde_fig,'deformstring','');
setappdata(pde_fig,'heightstring','');

% Solve PDE:
pdetool('solve')

MATLAB code for postprocessing and surface plot

To use this code snippet, the user must first export solution (\(u\)) and mesh data ([p e t]) from within the MATLAB pdetool window. The in the MATLAB command window, running the following function leads to a surface plot being created (example: figure 20):

```
pdeplot(p,e,t,'xydata',u,'zdata',u,'contour','off','colormap','jet')
```
% 'jet' is a colormap type. p e t are the mesh parameters points, ... 
elements and triangles while u is the solution. This solution data ... 
is unstructured and needs to be plotted using this pdeplot(...) ...
function.
Figure 20: Surface plot of temperature plotted using MATLAB’s *pdeplot(...)* function.

**MATLAB code interpolation**

The MATLAB pdeTool data is unstructured and needs to be properly passed through MATLAB’s internal functions so as to interpolate, say, temperature data at different points in the domain. The example code below allows for interpolating and finding the temperature at point $x = 0.5$ in the domain.

```
1 F = pdeInterpolant(p,t,u);
2 %Evaluate the interpolant at point, x=0.5
3 pOut = [0.5,0.5; 0.5,0.5];
4 uOut = evaluate(F,pOut)
```
Unsuccessful initial attempts of some students

Figure 21: Insufficient order of shape function (left) and improper use of boundary conditions (right).  

Figure 22: Second order terms of shape function missing (left) and improper use of boundary conditions (right).  

Figure 23: Incorrect sign in shape function (left) and improper use of boundary conditions (right).  

The unsuccessful first attempts at solving the heat transfer exam problem shows that students were not aware or had tenuous awareness of the effect of boundary conditions or choice of shape function, both being seminal components of the application of FEM (analytically or through software).