# Finite Element Methods and its Applications

For 2.S976
Introduction to Finite Element Methods for Mechanical Engineers
As taught by Prof. Anthony Patera in Spring 2019

Patrick Shin
MIT Mechanical Engineering

May 17, 2019

## Abstract

The finite element method is a numerical computation technique used extensively in engineering due to its ability to approximate complex solutions by breaking it down into repetitive computations. The mathematical foundations of the finite element method and physical scenarios where the it is implemented are discussed. The Rayleigh-Ritz method is the fundamental operation behind the methods developed in subsequent chapters. Two common engineering phenomena of interest, heat transfer and beam bending, are explored using the finite element method and demonstrate its utility in these scenarios.

# Contents

**1 The Rayleigh-Ritz Method 2**

**2 The FE Method for 1D 2nd-Order SPD BVPs 9**

**3 The FD-FE Method for the 1D Heat Equation:<br>Flipping Burgers 18**

**4 The FE-FE Method for 4th-Order BVPs (Bending): Xylophone 25**

# 1 The Rayleigh-Ritz Method

The Rayleigh-Ritz method is a mathematical process used to approximate a solution in the form of a summation of basis functions. This process is commonly used in finite element analysis because it can be systematically performed for each discretization in a simple or complex mesh to produce a result. Although invented in 1909 by Walther Ritz, it is only recently that its usage has grown: because of its repetitive nature, the Rayleigh-Ritz method is made practical in engineering by the raw computing power of modern technology.

The Rayleigh-Ritz method uses a governing partial differential equation with appropriate boundary conditions to derive the basis function coefficients. In this chapter, two specific types of boundary cases. Neumann-Robin (NR) and Dirichlet (D), are used to demonstrate the method. Note that $\kappa(x)$ is not the same as $\kappa$ used in the examples because it represents a physical property rather than a general function.

Consider the governing partial differential

$$-\frac{d}{dx}(\kappa(x)\frac{du}{dx}) + \mu(x)u = f_\Omega(x) \quad \text{in } \Omega \tag{1}$$

For the Neuman-Robin boundary condition example:

$$\begin{aligned} -\kappa\frac{d}{dx}(\pi R_0^2(1+\beta\frac{x}{L})^2\frac{du}{dx}) = 0 \quad \text{in } \Omega \\ \kappa\frac{du}{dx} = -q_1 \quad \text{on } \Gamma_1 \\ -\kappa\frac{du}{dx} = \eta_2(u-u_\infty) \quad \text{on } \Gamma_2 \\ \text{Where} \quad \kappa(x) = \kappa\pi R_0^2(1+\beta\frac{x}{L})^2 \end{aligned} \tag{2}$$

For the Dirichlet boundary condition example:

$$\begin{aligned} -\kappa A_{cs}\frac{d^2u}{dx^2} + \eta_3 P_{cs}u - \eta_3 P_{cs}u_\infty = 0 \quad \text{in } \Omega \\ u = u_{\Gamma_1} \quad \text{on } \Gamma_1 \\ -\kappa\frac{du}{dx} = 0 \quad \text{on } \Gamma_2 \\ \text{Where} \quad \kappa(x) = \kappa A_{cs} \end{aligned} \tag{3}$$

These can be manipulated to form the energy functional $\Pi$, which is minimized to find the basis function coefficients. The reasoning behind this step is that the energy functional of the real solution will always yield the minimum value. Any perturbations will cause the energy functional to increase and deviate from the local minima on either side, which is a clear indicator that it is diverging from the solution. Therefore, the minimum value of the energy functional derived from the boundary value problem and the selected basis functions will be the closest possible value to the true solution. As basis functions that are not the solution are added to the approximation, it is shown analytically and graphically that $\Pi$ is able to progressively better approximate the solution, further validating correct implementation.

The energy functional Π minimization for a Neumann-Robin boundary value problem:

$$\Pi(\omega) = \frac{1}{2}\int_0^L \kappa(x)(\frac{d\omega}{dx})^2 + \mu(x)\omega^2 dx + \frac{1}{2}(\gamma_1\omega^2(0) + \gamma_2\omega^2(L)) - \int_0^L f_\Omega(x)dx \tag{4}$$

The energy functional Π minimization for a Dirichlet boundary value problem

$$\Pi(\omega) = \frac{1}{2}\int_0^L \kappa(x)(\frac{d\omega}{dx})^2 + \mu(x)\omega^2 dx + \frac{1}{2}(\gamma_2\omega^2(L)) - \int_0^L f_\Omega(x)dx - \omega(L)f_{\Gamma_2} \tag{5}$$

But given the structure of these energy functionals, they can be easily deconstructed based on the dependency of a perturbation $\nu$ i.e. $\Pi(u+\nu)$. The only components that are linearly dependent of $\nu$ can be grouped together. This grouping is used in $Q(\alpha)$, defined by replacing u and $\nu$ in the grouping with the basis functions $\psi$ in question. Taking the derivative of Q with respect to $\alpha$ to find the minimum energy yields a system of linear equations

$$\underline{\beta\alpha^{RR}} == \underline{G} \tag{6}$$

Where the objective is to find only the alphas that minimize the energy. Matrices B and G are not identical for the Neumann-Robin and Dirichlet cases.

For the former:

$$\underline{B}_{i,j} = \int_0^L \kappa\left(x\right)\frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + \mu\left(x\right)\psi_i\psi_j dx + \gamma_1\psi_i\left(0\right)\psi_j\left(0\right) + \gamma_2\psi_i\left(L\right)\psi_j\left(L\right) \quad 1 \le i,j \le n^{RR} \tag{7}$$

$$\underline{G} = \int_0^L f_\Omega\left(x\right)\psi_i dx + f_{\Gamma_1}\psi_i\left(0\right) + f_{\Gamma_2}\psi_i\left(L\right), 1 \le i \le n^{RR} \tag{8}$$

For the latter

$$\underline{B}_{i,j} = \int_0^L \kappa\left(x\right)\frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + \mu\left(x\right)\psi_i\psi_j dx + \gamma_2\psi_i\left(L\right)\psi_j\left(L\right) \quad 1 \le i,j \le n^{RR} \tag{9}$$

$$\underline{G} = \int_0^L f_\Omega\left(x\right)\psi_i dx + f_{\Gamma_2}\psi_i\left(L\right) - u_{\Gamma_1}\underline{b} \quad 1 \le i \le n^{RR} \tag{10}$$

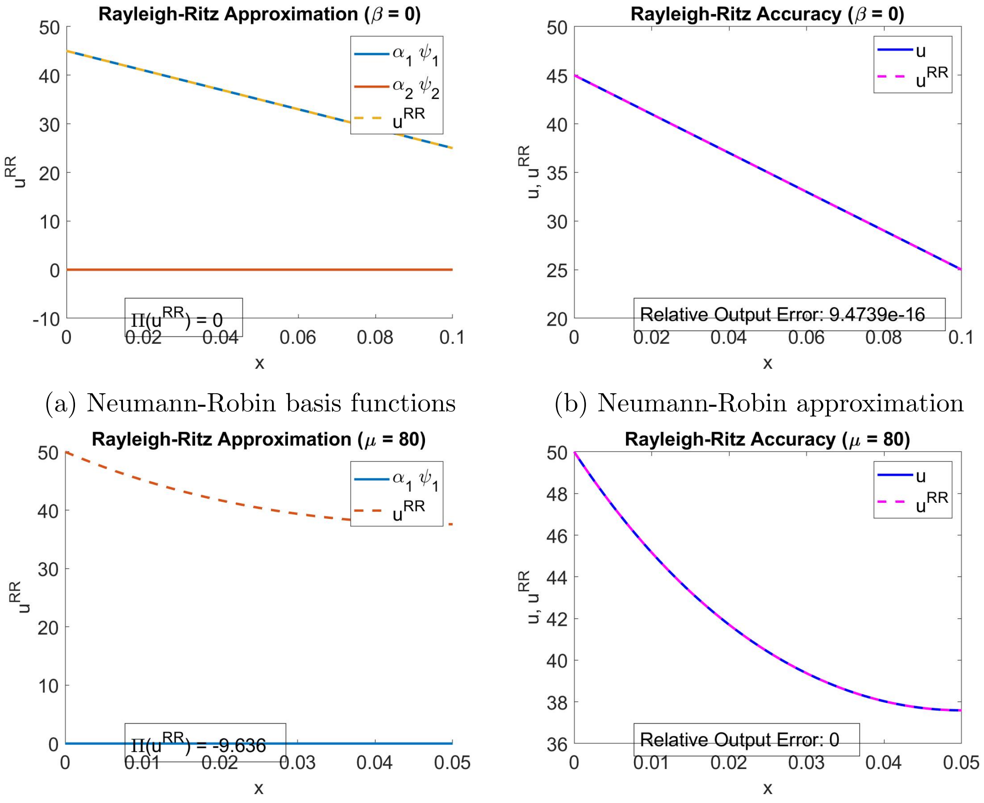
Where

$$\underline{b} = \int_0^L \kappa\left(x\right)\frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + \mu\left(x\right)\psi_i\psi_j dx + \gamma_2\psi_i\left(L\right)\psi_j\left(L\right) \quad 1 \le i \le n^{RR},\ j = 0 \tag{11}$$

$$\underline{\widetilde{\propto^{RR}}} = \begin{bmatrix}\propto_0^{RR} & \underline{\propto^{RR}}\end{bmatrix}^T \tag{12}$$

By solving for $\alpha^{RR}$, the Rayleigh-Ritz approximations can successfully be used to determine a solution. The specific Rayleigh-Ritz equation differ between the two cases being followed, but they both consist of the summation of basis functions factored by the Rayleigh-Ritz coefficients. Matlab was used to calculate and plot the results of these approximations across various cases.

Neumann-Robin



(a) Neumann-Robin basis functions

(b) Neumann-Robin approximation

(c) Dirichlet basis functions

(d) Dirichlet approximation

Figure 1: Basis functions and approximations for Neumann-Robin and Dirichlet conditions.

$$u^{RR}\left(x\right) = \sum_{i=1}^{n^{RR}} \alpha_i^{RR}\psi_i\left(x\right) \tag{13}$$

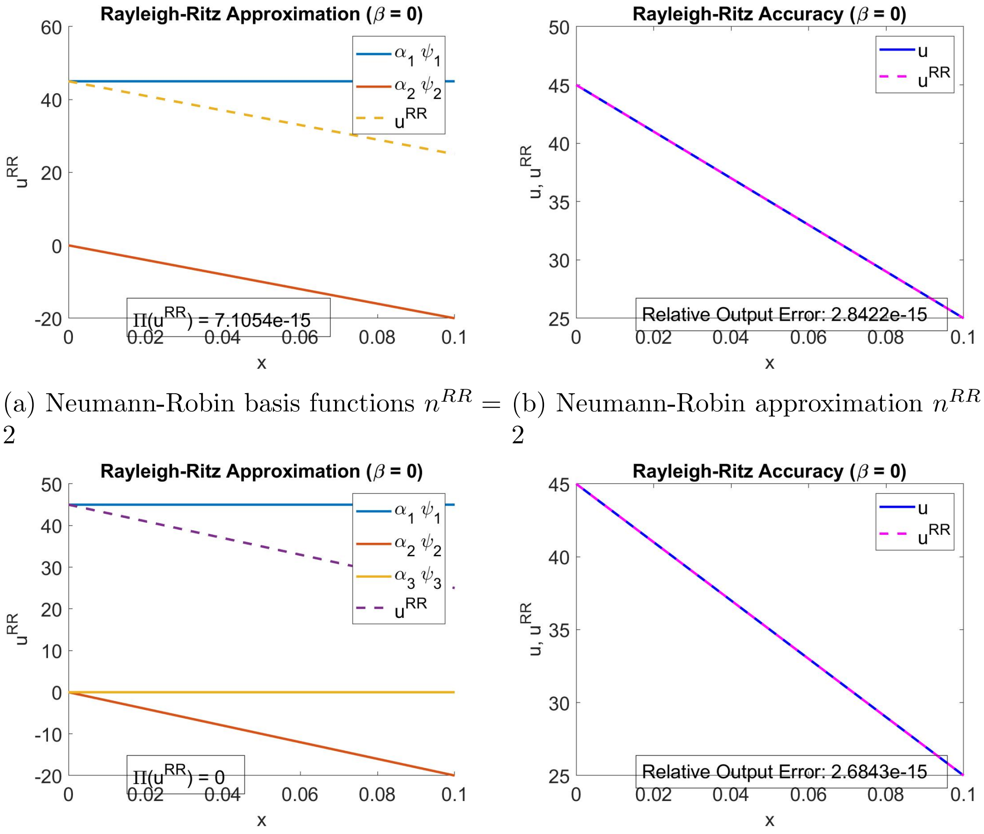
Dirichlet

$$u^{RR}\left(x\right) = u_{\Gamma_1}\psi_0\left(x\right) + \sum_{i=1}^{n^{RR}} \alpha_i^{RR}\psi_i\left(x\right) \tag{14}$$

First, to verify the functionality of the program, the exact solution was included as a basis function along with a linear function. As expected, the linear basis function received a Rayleigh-Ritz coefficient of zero, and the exact solution received a non-zero coefficient. This aligns with the Minimization Proposition which states that the energy functional of the real solution with a nonzero perturbation added will be greater than the energy functional of only the real solution.

Because the zeroth basis function is included in the Rayleigh-Ritz approximation outside of the summation, it was not plotted as a basis function but is still contributing to the calculation as shown above.

After the program has been verified, the basis functions can be changed to investigate the effectiveness of the Rayleigh-Ritz method for these systems. The following basis functions were tested:



(a) Neumann-Robin basis functions $n^{RR} = 2$

(b) Neumann-Robin approximation $n^{RR} = 2$

(c) Neumann-Robin basis functions $n^{RR} = 3$

(d) Neumann-Robin approximation $n^{RR} = 3$

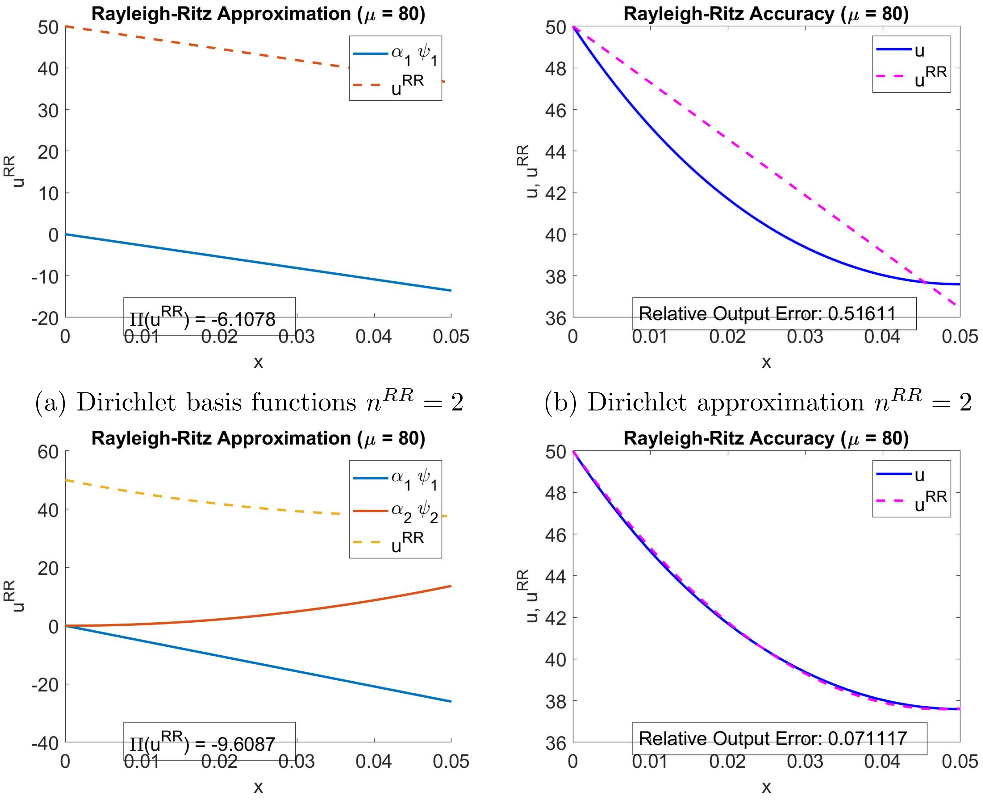
Figure 2: Basis functions and approximations for Neumann-Robin condition of different basis function quantities.

$$\text{Neumann-Robin} \quad \psi_1(x) = 1, \psi_2(x) = x, \psi_3(x) = x^2 \tag{15}$$

$$\text{Dirichlet} \quad \psi_0(x) = 1, \psi_1(x) = x, \psi_2(x) = x^2 \tag{16}$$

In three out of the four Rayleigh-Ritz accuracy figures shown above, the program was able to approximate the solution reasonably well. It successfully attributed zero coefficients to exponential basis functions when the solution was a linear relationship. The exception was the third case where the program tried its best to fit a straight line onto a curved one, and managed to get the start and end points fairly close to the solution values. The results of these two tests corroborate the proper functionality of the program.

A caveat with the Rayleigh-Ritz approximation that must be considered is the change in error that can occur at different scales and parameters. For instance, the differential equations governing the boundary value problem may only hold true for a certain range of common physical parameters, or in the case of the Dirichlet case, it can correspond with different physical processes (natural convection/radiation versus forced convection versus change-of-phase). Therefore, not all



(a) Dirichlet basis functions $n^{RR} = 2$

(b) Dirichlet approximation $n^{RR} = 2$

(c) Dirichlet basis functions $n^{RR} = 3$

(d) Dirichlet approximation $n^{RR} = 3$

Figure 3: Basis functions and approximations for Dirichlet condition of different basis function quantities.

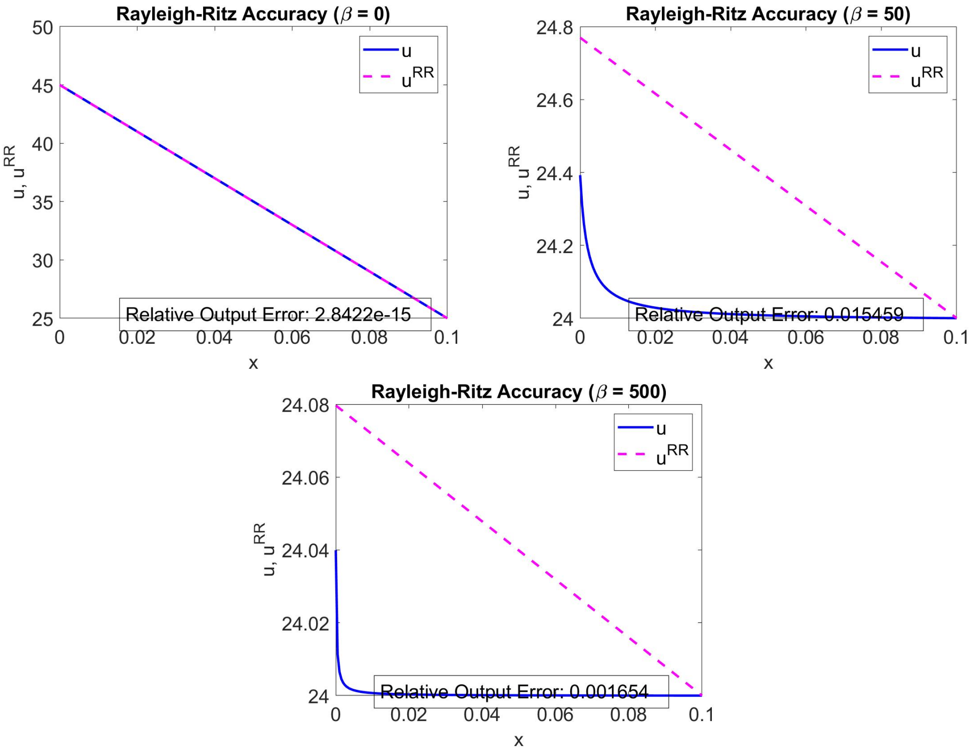


Figure 4: Rayleigh-Ritz accuracies for the Neumann-Robin case.

Rayleigh-Ritz setups work with all scenarios, and it is up to the engineer to analyze the results and set up the problem appropriately.

For the Neumann-Robin case, it was found that as $\beta$ increased, the curvature of the exact solution increased, which made it difficult for the program to fit the basis functions and increased the difference between the approximation and the exact solution.

For the Dirichlet case, a similar behavior was observed: as $\eta_3$ was increased, the error between the approximation and the exact solution also increased. This is due to the increased curvature of the exact solution, which made the combinations of basis function inadequate to fit the curve. As shown in these two examples, an exact solution which behaves as a steep curve followed by a plateau is difficult to approximate using the Rayleigh-Ritz method.

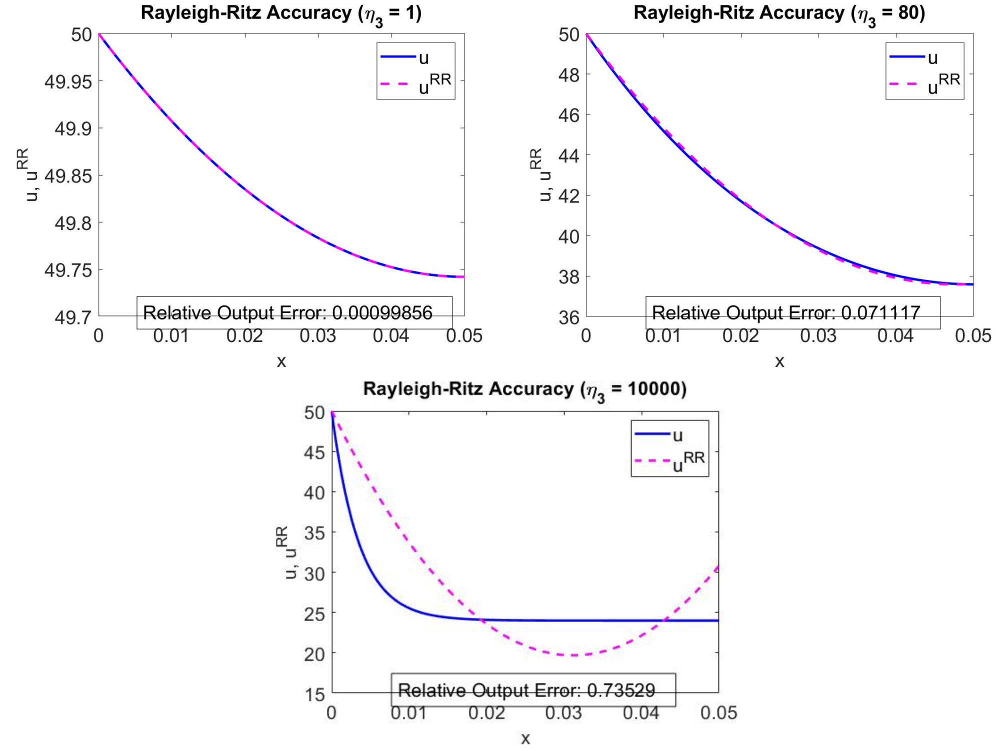


Figure 5: Rayleigh-Ritz accuracies for the Dirichlet case.

## 2 The FE Method for 1D 2nd-Order SPD BVPs

The finite element (FE) method implements the Rayleigh Ritz technique by dividing the domain into a series of elements that form a mesh. The basis functions that reside in these elements are used in a similar fashion to estimate the true solution. The ability to create these meshes allows solutions that could not be estimated in a single Rayleigh Ritz element to be accurately found. Equidistant nodes (or not, in the adaptive case) are used to define the boundaries of each element, and the nodal basis functions within can be based on linear or quadratic functions.

Three models are explored in this chapter - two of which are taken from the previous chapter. Model I describes heat conduction in a conical frustum with insulated lateral surfaces and heat flux boundary condition on the left face and a heat transfer coefficient boundary condition on the right (Neumann-Robin boundary conditions). Model II describes a cylindrical thermal fin with an imposed temperature on the left surface (Dirichlet boundary condition) and a zero flux boundary condition on the right. Model III is similar to Model I, in that it describes heat conduction through a slab with insulated lateral surfaces but with heat transfer coefficient boundary conditions on both surfaces. The dual Neumann-Robin boundary conditions in Model III could physically represent a wall of a building in a winter storm where different heat transfer coefficients and ambient temperatures exist on both sides. Additionally, Model II is better suited to test the heat transfer coefficient boundary condition because it is nonzero at its left boundary.

Consider the governing partial differential

$$-\frac{d}{dx}(\kappa(x)\frac{du}{dx}) + \mu(x)u = f_\Omega(x) \quad \text{in } \Omega \tag{17}$$

Incorporating the corresponding variables for Model I gives

$$\begin{aligned} -\kappa\frac{d}{dx}(\pi R_0^2(1+\beta\frac{x}{L})^2\frac{du}{dx}) &= 0 \quad \text{in } \Omega \\ \kappa\frac{du}{dx} &= -q_1 \quad \text{on } \Gamma_1 \\ -\kappa\frac{du}{dx} &= \eta_2(u-u_\infty) \quad \text{on } \Gamma_2 \end{aligned} \tag{18}$$

With the output as the temperature at $x = 0$

$$s \equiv u(0) \tag{19}$$

For Model II

$$\begin{aligned} -\kappa A_{cs}\frac{d^2u}{dx^2} + \eta_3 P_{cs}u - \eta_3 P_{cs}u_\infty &= 0 \quad \text{in } \Omega \\ u &= u_{\Gamma_1} \quad \text{on } \Gamma_1 \\ -\kappa\frac{du}{dx} &= 0 \quad \text{on } \Gamma_2 \end{aligned} \tag{20}$$

With the output

$$s \equiv -\kappa\frac{du}{dx}(x-0) \tag{21}$$

For Model III

$$\begin{aligned} -\frac{d}{dx}(\kappa A_{cs}\frac{du}{dx}) &= 0 \quad \text{in } \Omega \\ \kappa\frac{du}{dx} &= \eta_1(u-u_{\infty\Gamma_1})\text{on } \Gamma_1 \\ -\kappa\frac{du}{dx} &= \eta_2(u-u_{\infty\Gamma_2}) \quad \text{on } \Gamma_2 \end{aligned} \tag{22}$$

With the same output as Model I

$$s \equiv u(0) \tag{23}$$

The finite element method begins the same way as the Rayleigh Ritz method in the previous chapter: the governing equations are manipulated using the energy minimization principle to derive the energy functional. This equation can then be used to form a relationship that estimates the solution through a linear combination of basis functions. In the $P_1$ case, there are two linear nodal basis functions within each element that form an X. Because every nodal function spans two elements and reaches its maximum at the center node while the other two functions are zero, the coefficient of that nodal function is a valid point in the solution. These points can be connected to form a piecewise-linear, continuous function that estimates the solution. Likewise in the $P_2$ case, there are now three nodal basis functions in each element, all of which are quadratic functions. Each nodal basis function now overlaps with at most five other neighboring functions. The same principle applies: at the maxima of each function, all others are zero. Because these functions overlap, unique combinations can be computed to approximate a wide range of curves based on the basis functions available. Therefore the approximate solution can be represented as follows:

$$u_h(x) = \sum_{i=1}^{n} u_{hi}\varphi(x) \tag{24}$$

The energy functional can be rewritten in matrix form in order to isolate the solution vector

$$\underline{A} \quad \underline{u_h} = \underline{F} \tag{25}$$

$$A_{ij} = \int_0^L \kappa(x)\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \mu(x)\varphi_i\varphi_j dx + \gamma_1\varphi_i(0)\varphi_j(0) + \gamma_2\varphi_i(L)\varphi_j(L) \tag{26}$$

$$F_i = \int_0^L f_\Omega(x)\varphi dx + f_{\Gamma_1}\varphi_i(0) + f_{\Gamma_2}\varphi_i(L) \tag{27}$$

Therefore, for Neumann-Robin boundary conditions, such as Model I, the matrices become

$$A_{ij} = \int_0^L \kappa\pi R_0^2(1+\beta\frac{x}{L})\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \eta_2\pi R_0^2(1+\beta\frac{x}{L})\varphi_i(L)\varphi_j(L) \tag{28}$$

$$F_i = \int_0^L q_1\pi R_0^2\varphi_i(0) + \eta_2\pi R_0^2 u_\infty(1+\beta)^2\varphi_i(L) \tag{29}$$

And for Model III

$$A_{ij} = \int_0^L \kappa A_{cs}\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \eta_1 A_{cs}\varphi_i(0)\varphi_j(0) + \eta_2 A_{cs}\varphi_i(L)\varphi_j(L) \tag{30}$$

$$F_i = \int_0^L u_{\infty_1} A_{cs}\eta_1\varphi_i(0) + u_{\infty_2} A_{cs}\eta_2\varphi_i(L) \tag{31}$$

For Dirichlet boundary conditions, these matrices differ slightly

$$u_h(x) = u_{\Gamma_1}\varphi_1(x) + \sum_{i=2}^{n} u_{hi}\varphi(x) \tag{32}$$

$$\widetilde{A_{ij}} = \int_0^L \kappa(x)\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \mu(x)\varphi_i\varphi_j dx + \gamma_1\varphi_i(0)\varphi_j(0) + \gamma_2\varphi_i(L)\varphi_j(L) \tag{33}$$

$$\widetilde{F_i} = \int_0^L f_\Omega(x)\varphi dx + f_{\Gamma_1}\varphi_i(0) + f_{\Gamma_2}\varphi_i(L) \tag{34}$$

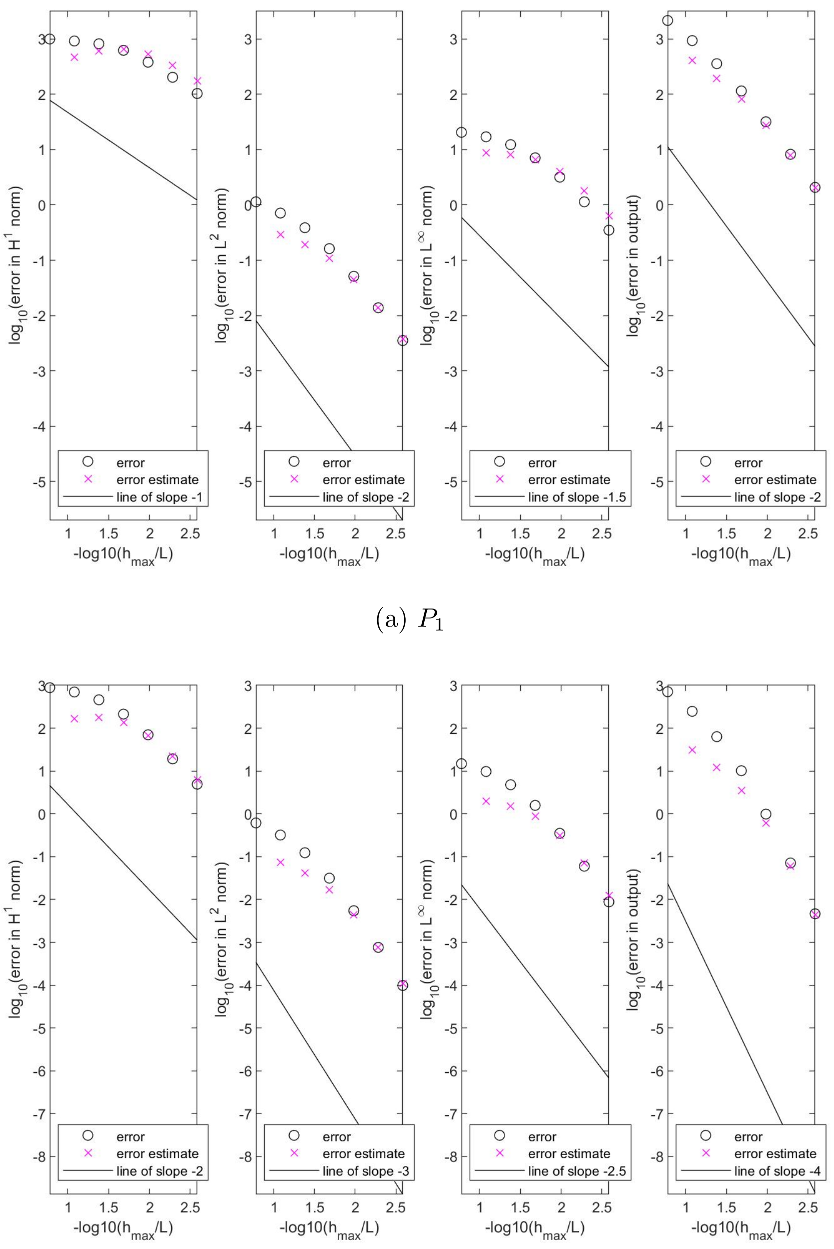
Where $\underline{A}, \underline{u_h^0}$ and $\underline{F}$ are subsets of $\underline{\widetilde{A}}, \underline{u_h}$ and $\underline{\widetilde{F}}$, respectively. These subsets can be used to find the solution

$$\underline{A} \quad \underline{u_h^0} = \underline{F} - u_{\Gamma_1}\underline{b} \tag{35}$$

The implementation of these integrals involve the use of quadratures to perform numerical integration on elemental matrices instead. Of course, this method has some finite precision, and its performance can be observed though convergence plots. For most systems, a decrease in error is expected as the mesh is refined due to the higher resolution. The slopes of the black lines correspond to the exponents of the element size when calculating the error bounds for the norms and outputs and is a function of P. If the plots converge at this rate, the you know that the error is decreasing proportional to the element size. The figure below shows the proper convergence of Model II from $u_h$ to $u$. The parallel convergence rate indicates that the numerical integration is benefitting from a denser mesh as expected. The finite method is effective because of its ability to break down complex problems into small, discrete elements that can then be solved using the same method, taking advantage of modern computing power. For instance, the sparse nature of the matrices involved in the calculations allow the computational cost to be optimized.

Although convergence at the correct rate is a strong indicator for the validity of a finite element method script, it is not a guarantee. The method of manufactured solutions allows the true solution to be manually interrogated. By solving for the solution algebraically using the given functions and variables in the governing equations and boundary conditions, it can be used to calculate the error. To further verify the numerical integration implementation, a variation on Model II can be used. By substituting $\mu(x)$ with a linear function instead of a constant and then using an assumed quadratic solution to solve for $f_\Omega$ and $f_{\Gamma_2}$, a correct implementation should still yield convergence in this case.

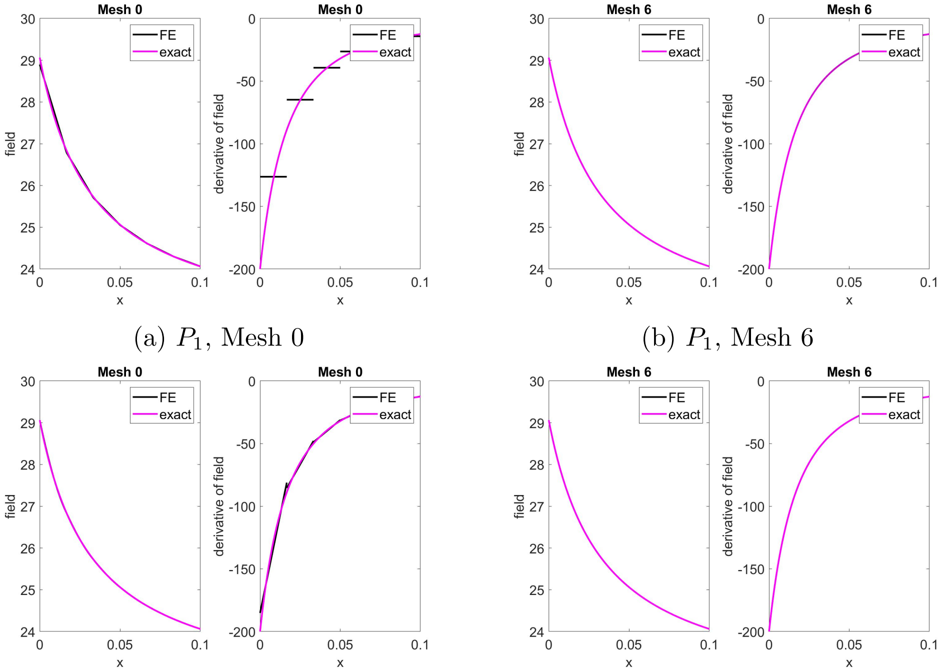
In addition to proper implementation of numerical integration, the proper implementation of boundary conditions is also critical. Models I and III support the functionality of the latter because they rely on it for the correct estimations of the solution. The mesh and convergence plots of Model I in figures 7 and 8, respectively, show the decrease in error as it is uniformly refined at the correct rate. Without proper boundary condition implementation, the error estimate and the error markers would not coincide.



(a) $P_1$

(b) $P_2$

Figure 6: Convergence plots for Model II $P_1$ and $P_2$ cases.



(a) $P_1$, Mesh 0

(b) $P_1$, Mesh 6

(c) $P_2$, Mesh 0

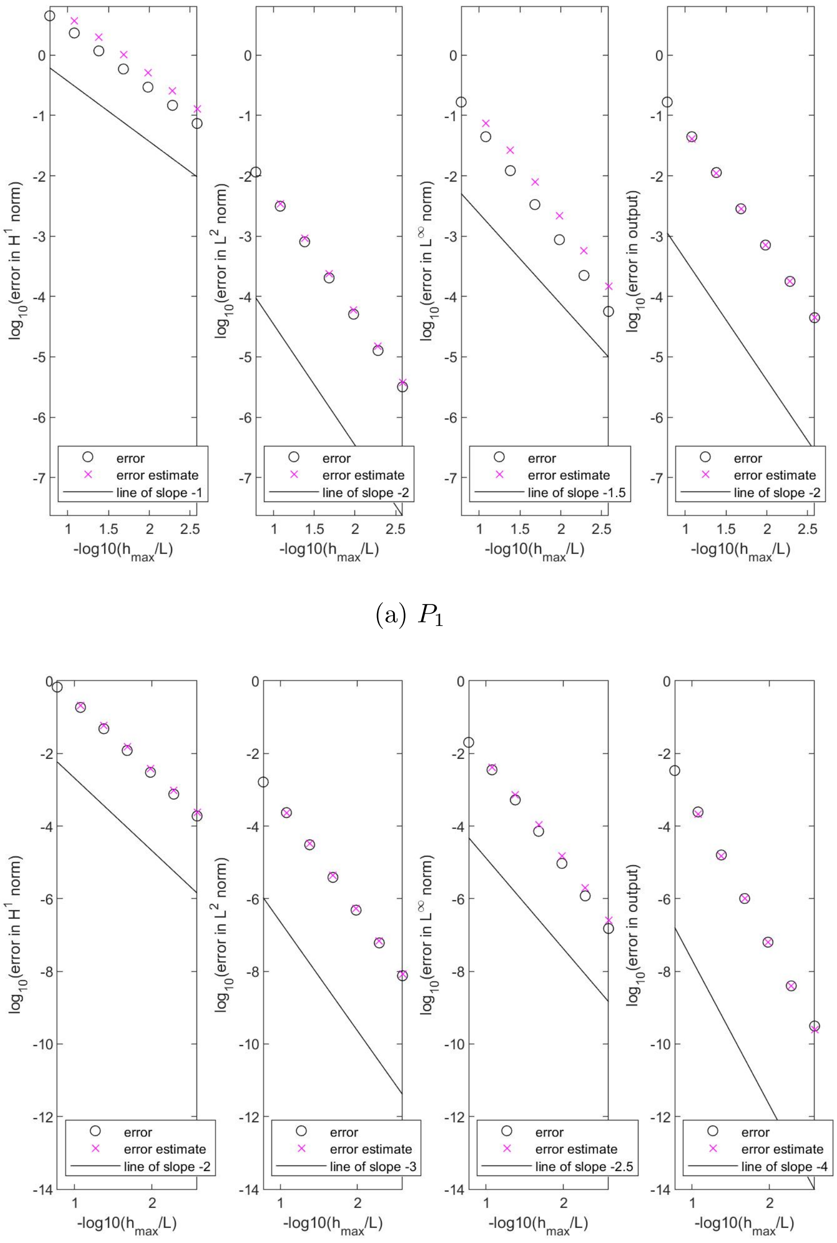
(d) $P_2$, Mesh 6

Figure 7: First and final field and derivative of field plots for Model I $P_1$ and $P_2$ cases.

Because Model III has insulated lateral surfaces ($\mu(x) = 0$), the solution is exactly linear. Therefore, when the numerical integration occurs, very small error should occur. As seen in Figure 9, the fields show agreement with the exact solution, but the derivative fields vary a marginal amount. This behavior is translated into the convergence plots in Figure 10, where the errors do not converge. This is due to the finite precision of the computer causing amplification of round-off error. Although it does not converge, the errors are significantly smaller than that of the other two models. In the $P_2$ case this behavior is more apparent, as it is more difficult to approximate a linear solution with quadratic functions. Because of the linear nature of this model's solution, it provides greater implementation confidence supported by the very small and noisy errors.

Take a hypothetical Model X, for which we assume we do not know the exact solution to. For a fine mesh, the extrapolation error estimates are observed to converge at correct rates across all norms. In most cases it can be assumed that the estimated solution $u_h$ is converging to the exact solution $u$. But because the exact solution is not known, there is a chance that the estimated solution is converging to an incorrect solution. This incorrect solution could be a very similar function but perhaps offset from the exact solution, causing the convergence to appear correct. This can be caused by incorrect parameters or a false derivation of the exact solution.

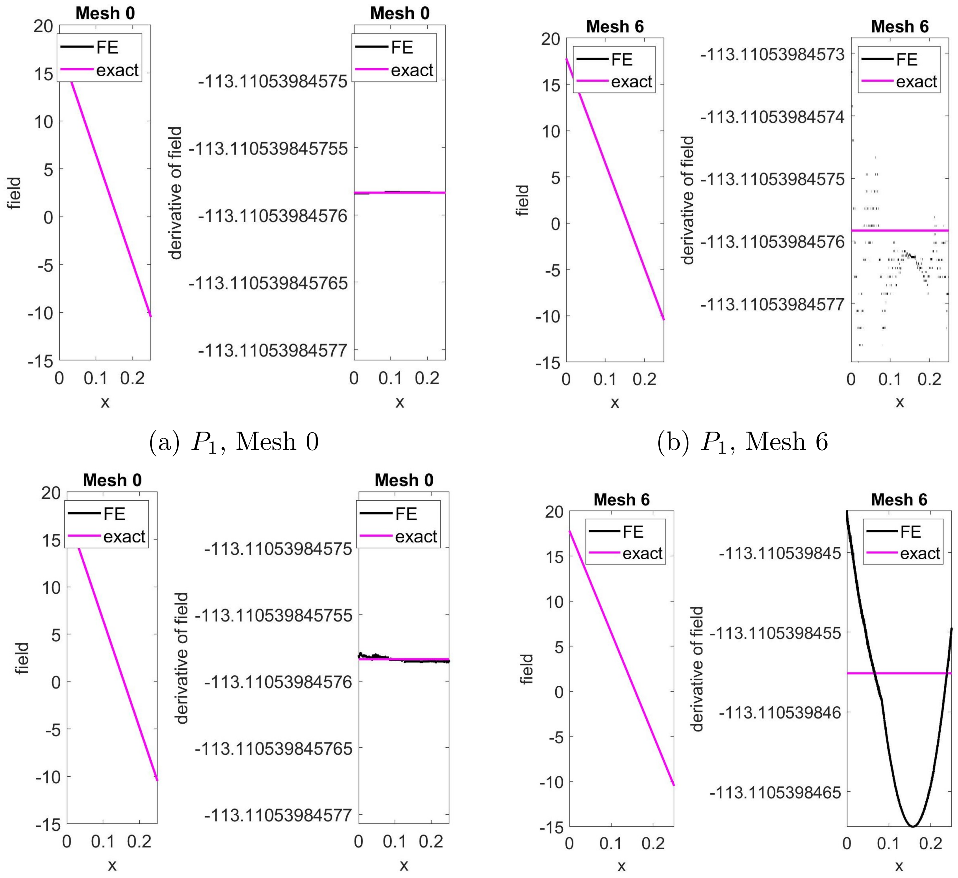
In order to explore the behavior of accuracy and numerical specification, a certain number of refinements can be imposed. By forcing Model II to undergo 9 meshes in the $P_1$ case without an exact solution, it can be seen in figure 12 that mesh 7 is the coarsest mesh such that $||u - u_h||_{L_\infty(\Omega)} \leq 1.00$. By extracting the data point from the figure, an upper bound of $||u - u_h||_{L_\infty(\Omega)} \leq 0.636$ is possible. For the error in the output of mesh 5, an upper bound $|s - s_h| \leq 27.733$ is seen. When plotted with the errors for the exact solution, it can be seen that the errors of the $L_\infty$ norm and



(a) $P_1$

(b) $P_2$

Figure 8: Convergence for Model I $P_1$ and $P_2$ cases.



(a) $P_1$, Mesh 0

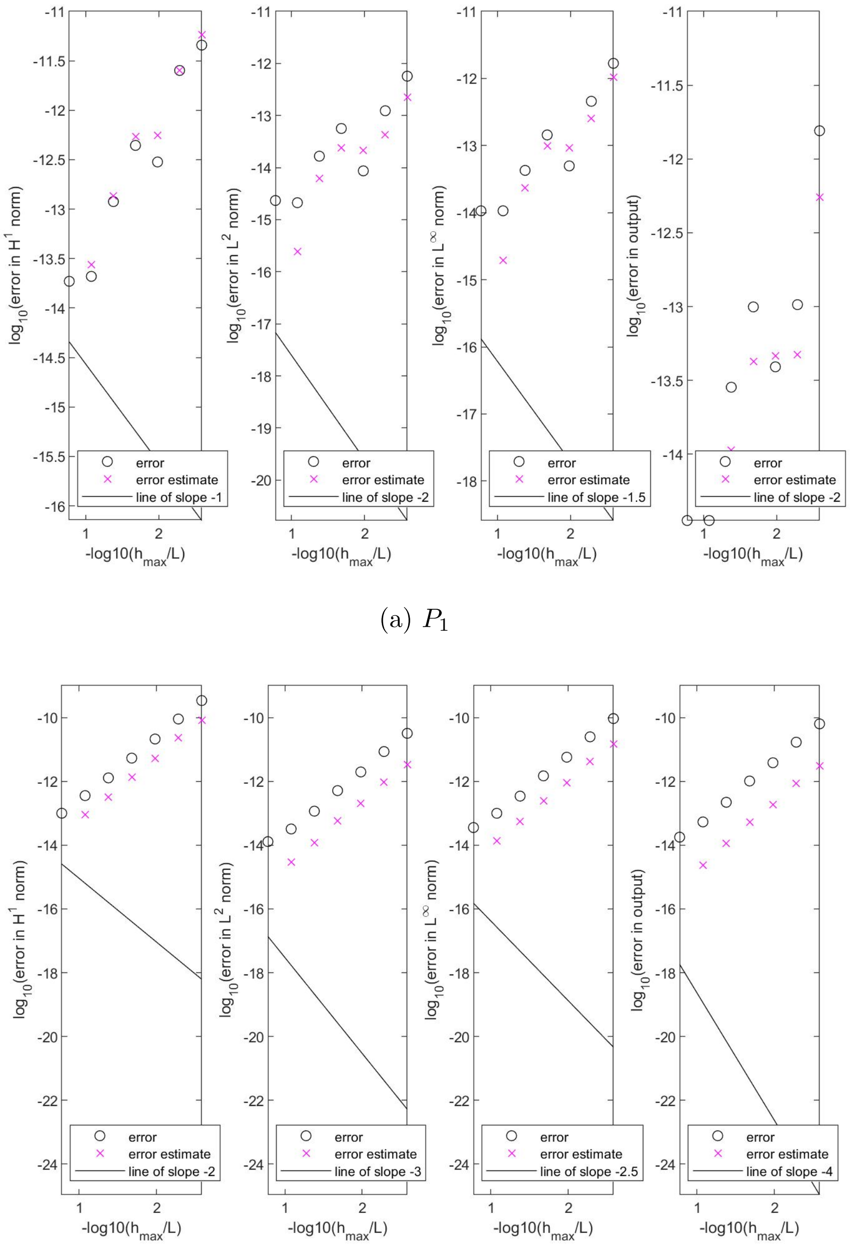
(b) $P_1$, Mesh 6

(c) $P_2$, Mesh 0

(d) $P_2$, Mesh 6

Figure 9: First and final field and derivative of field plots for Model III $P_1$ and $P_2$ cases.

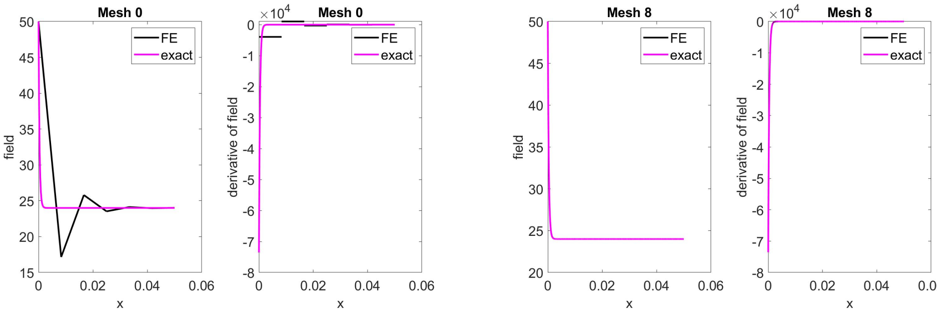
the output with respect to the exact solution have less than or equal error bounds.



(a) $P_1$

(b) $P_2$

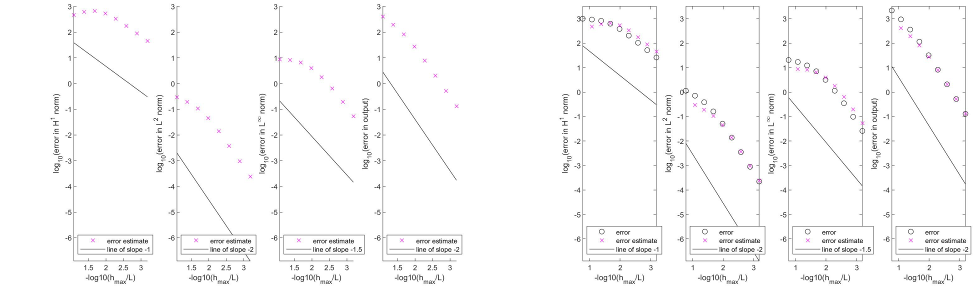
Figure 10: Convergence for Model III $P_1$ and $P_2$ cases.



(a) $P_1$, Mesh 0

(b) $P_1$, Mesh 8

Figure 11: First and final field and derivative of field plots for Model II $P_1$ case.



(a) Without exact solution

(b) With exact solution

Figure 12: Convergence for Model II $P_1$ case with and without exact solution.

# 3 The FD-FE Method for the 1D Heat Equation: Flipping Burgers

In engineering, a common subject of interest is the time-dependent behavior of a system. This chapter builds upon the existing knowledge of FE methods to formulate a version capable of time-dependent analysis. Like the previous chapters, a Neumann-Robin/Neumann-Robin boundary condition case will serve as the starting point of discussion. The high-level techniques used to approximate a solution consist of taking time derivatives of the governing equations that form a relationship of matrices consisting of systems of ODE's in time. Then, the finite difference (FD) method is used to approximate the solutions to these differential equations. The conversion of these initially non-linear ODE's into a linear system allows computationally efficient matrix algebra methods to be used to approach a solution.

The Neumann-Robin/Neumann-Robin boundary condition model takes a general form that is similar to the ones used for the steady cases, with an additional term to account for the time derivative of the temperature.
Consider the governing partial differential

$$-\frac{d}{dx}(\kappa(x)\frac{du}{dx}) + \mu(x)u = f_\Omega(x) - \rho(x)\dot{u} \quad \text{in } \Omega \tag{36}$$

With boundary conditions

$$\begin{aligned} \kappa\frac{du}{dx} &= \gamma_1 - f_{\gamma_1} \quad \text{on } \Gamma_1 \\ -\kappa\frac{du}{dx} &= \gamma_2 - f_{\gamma_2} \quad \text{on } \Gamma_2 \end{aligned} \tag{37}$$

Manipulating these equations to take a form compatible with the finite difference method requires a few extra steps compared to the steady state example. First, the $u$ and $\dot{u}$ functions are defined as summations of output and basis function vectors. Next, the A and F matrix relationship that result from using the steady state case are not ideal: the F matrix in particular is a function of $\dot{u}$ which makes it difficult to solve numerically. This is where a new matrix called the inertial mass matrix ($\underline{M}^{inertia}$) is necessary. This is used to separate the $\dot{u}$ components from the incompatible F matrix into an isolated term after the F matrix has been expanded.

$$\begin{aligned} u_h(x,t) &= \sum_{j=1}^{n} u_{hj}(t)\varphi_j(x) \\ \dot{u}_h(x,t) &= \sum_{j=1}^{n} \dot{u}_{hj}(t)\varphi_j(x) \end{aligned} \tag{38}$$

$$\underline{M}_{i,j}^{inertia} = \int_0^L \rho(x)\varphi_i\varphi_j dx \quad 1 \leq i,j \leq n \tag{39}$$

The resulting matrix expression consists of isolated $u$ and $\dot{u}$ terms as well as computationally compatible matrices.

$$\underline{M}^{inertia}\underline{\dot{u}}_h + \underline{A}\underline{u}_h = \underline{F} \tag{40}$$

From here, the finite difference method is used to substitute the $u_h$ and $u\dot{)}h$ terms with expressions that relate the output at the current time step with the previous one. This is where a new variable that determines the time step size is incorporated. The substitution for the $u_h$ terms has three variations based on what approximation scheme is used. Based on the definition of the $\theta$ variable, either the Euler forward, Euler backward, or Crank-Nicholson methods can be used by setting $\theta = 0, 1, \frac{1}{2}$, respectively.

$$\dot{\underline{u}}_h \approx \frac{\underline{u}^k_{h,\Delta t} - \underline{u}^{k-1}_{h,\Delta t}}{\Delta t} \tag{41}$$

$$\underline{u}_h \approx \theta \underline{u}^k_{h,\Delta t} + (1-\theta)\underline{u}^{k-1}_{h,\Delta t} \tag{42}$$

Performing the finite difference substitution and rearranging terms yields the following expression:

$$\underline{u}^k_{h,\Delta t} = (I_h u_{ic}) \quad \text{for} \quad k = 1 \tag{43}$$

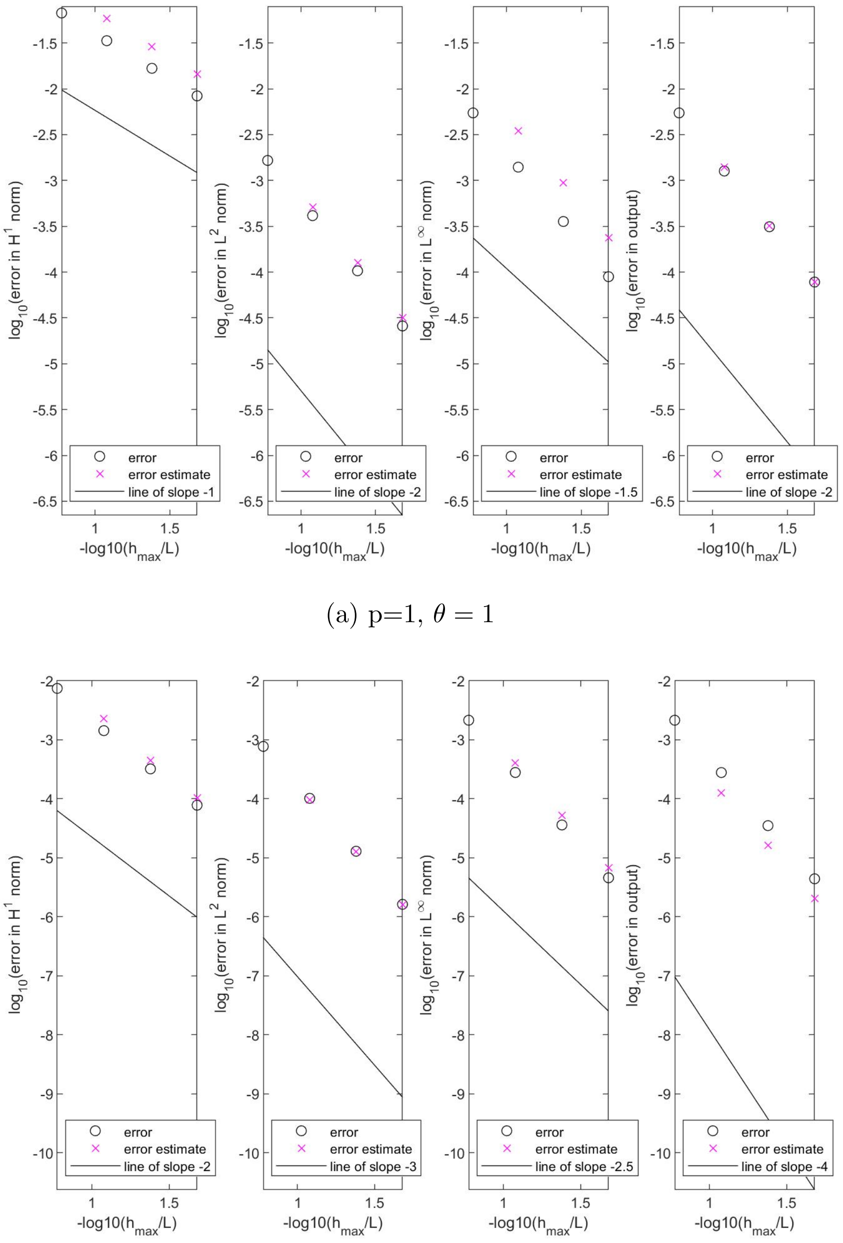
$$(\frac{\underline{M}^{inertia}}{\Delta t} + \theta \underline{A})\underline{u}^k_{h,\Delta t} = (\frac{\underline{M}^{inertia}}{\Delta t} - (1-\theta)\underline{A})\underline{u}^{k-1}_{h,\Delta t} + \underline{F} \quad \text{for} \quad k = 2 : n_{tsteps} \tag{44}$$

In order to validate correct implementation of this method, a test case modeling the temperature of a semi-infinite fin was used. The $L^2(\Omega)$ norm is an ideal indicator to validate correct implementation through convergence because its definition results in the optimal balance of spatial convergence rates within combinations of $p = 1, 2$ and $\theta = 1, \frac{1}{2}$. As shown, the convergence of the $L^2(\Omega)$ norm thats follows the correct slope and the alignment of the true errors with the error estimate suggests a correct implementation of code. When investigating the convergence behavior for uniform refinements, a choice of $\sigma$ is selected based on the values of P and $\theta$ (related to the finite element parameters and estimation method of choice) being used. Each norm has a different matrix of $\sigma$ values that determine its optimal behavior and therefore the proper rate of convergence to expect.

A more complex implementation of this method models the temperature of a burger patty during a three stage cooking process. By having three discrete time frames in which the system behavior is the same but different boundary conditions and variables, the cooking, flipping, and cooling of a burger patty is explored. The Matlab flip command can be used to quite literally flip the system numerically so that the program understands which side is facing where. This allows the system state to be remembered and carried onto the next process time window, similar to a checkpoint operation.

Observing the program outputs confirms correct implementation through the convergence plots of the first process time window and the intuitive agreement with the temperature profile plots. THe convergence plot is only for the first process time window because since the other two processes are programmed almost identically, the convergence behavior for the first window is enough to understand that of the other two.

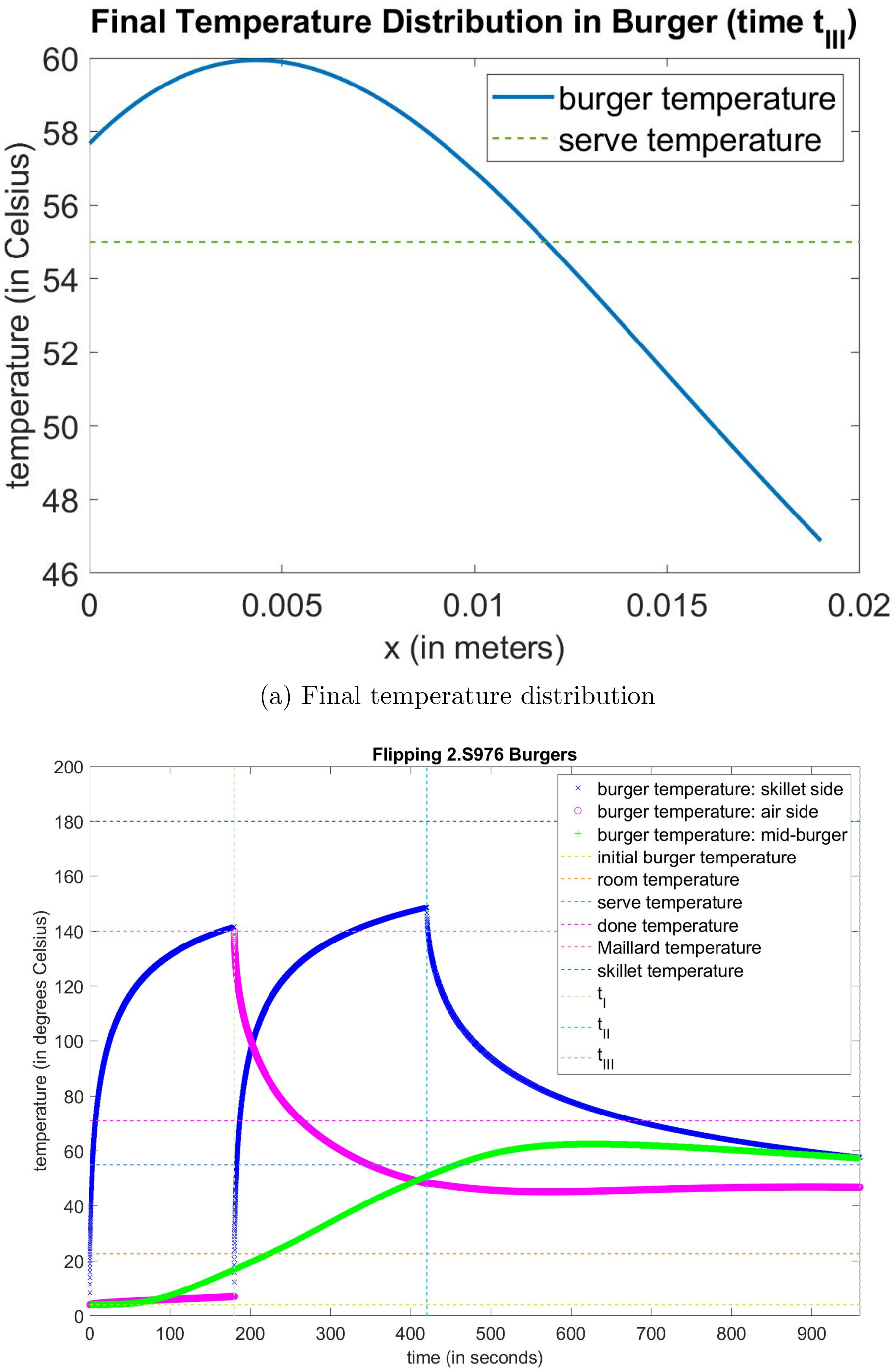
Good convergence and intuitive agreement does not guarantee an efficient program. Two different approaches $[p = 1, \theta = 1]$ (Euler Backward) and $[p = 2, \theta = \frac{1}{2}]$ (Crank-Nicholson) were used to explore the effect of error tolerance on these values. As shown in the two convergence graphs, the coarsest FE mesh required to achieve an error less than 0.001°C are significantly different. For the first approach, 5 uniform refinements were required to achieve an error of $3.6 \times 10^{-4}$ using a mesh size of $9.9 \times 10^{-5}$ and. For the second approach, two refinements are needed to achieve and



(a) p=1, $\theta = 1$

(b) p=2, $\theta = \frac{1}{2}$

Figure 13: Convergence plots for semi-infinite test case.



(a) Final temperature distribution

(b) Temperature plot

Figure 14: Convergence plot and temperature plots of a burger patty model.

error of $1.38 \times 10^{-4}$ using a mesh size of $7.92 \times 10^{-4}$. This is because for the $p = 2$ cases, the error value needs to be doubled due to the nature of the model, therefore causing a discrete y-shift upwards in the error graph.

It can be assumed that the computational time is approximately proportional to the number of computational operations, so a ratio of the operational quanitites can be taken to understand the difference in efficiencies between the two approaches. The product of the number of timesteps, number of elements, and the p-value is used to represent this quantity.

$$n_{tsteps} = n_{init} * 2^l \quad n_{el} = tstep_{init} * \sigma^l \tag{45}$$

$$O \approx n_{tsteps} * (n_{el} * p) \tag{46}$$

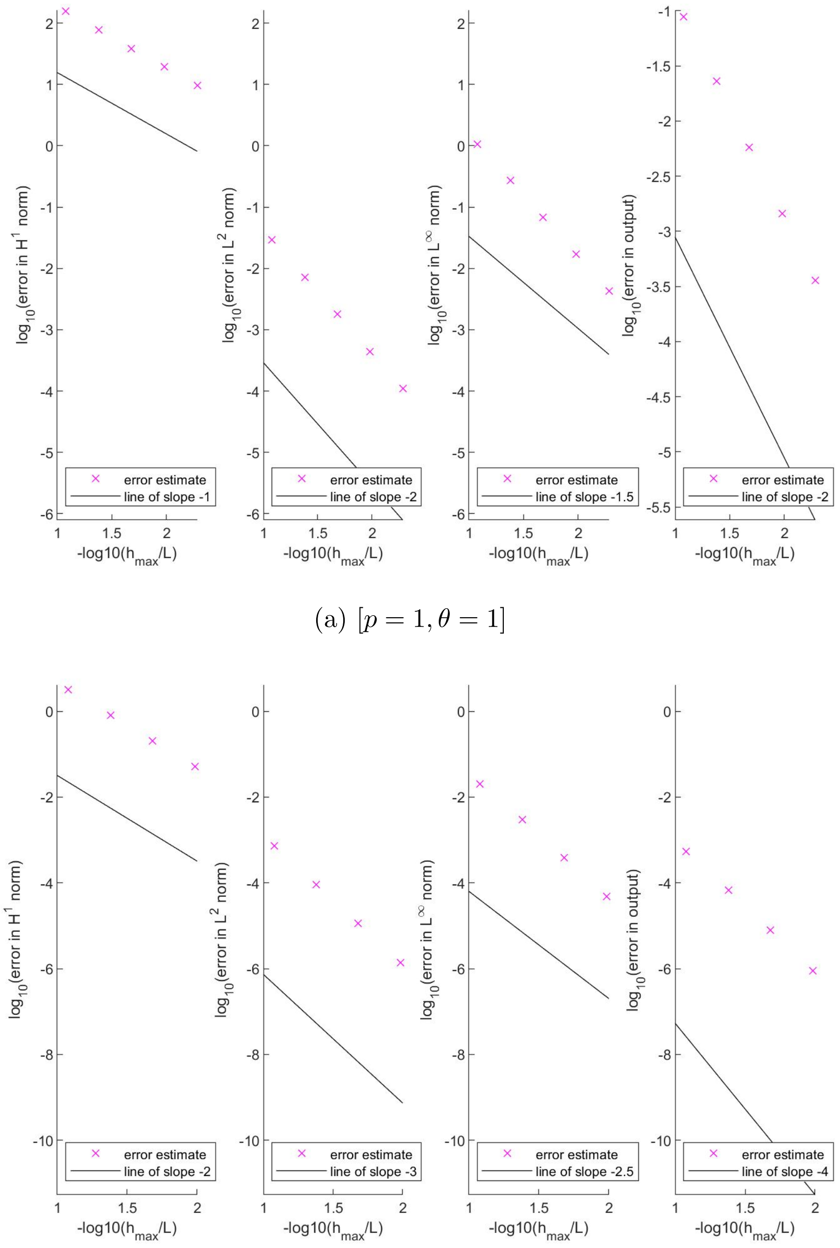
THis expression is further simplified by taking the ratio which cancels out the common variables which leaves only the variables with number of refinements as exponents. Plugging in the corresponding values for each approach yields an operational ratio for $[p = 1, \theta = 1]$ relative to $[p = 2, \theta = \frac{1}{2}]$ of 512. For the p=2 case, the quantity was doubled because it was assumed that solving a penta-diagonal matrix requires double the operational count compared to a tri-diagonal matrix. This states that the latter approach is superior in terms of efficiency.

The parameters of the burger model can be modified to understand if the results make sense for different recipes. Three sources of online information suggested new parameters for the hamburger dimensions [1], skillet temperature, and cooking times[2][3]. The burger diameter was increased to 4.5 inches (the ”perfect” burger diameter according to Adjust-A-Burger), the skillet temperature was increased to 375°F (190°C), the cooking times for each side became 4 min, and the resting time became 5 minutes. As shown in the new temperature time profiles, the model agrees in several ways with the expected results. First, the shape of the profiles is similar to that of the initial setup, implying that the behavior of the system has not changed dramatically. Second, the recipe calls for an internal temperature of 160°F (71°C) for safety, and the plot of mid-burger temperature settles to precisely that temperature at the end of the cooling phase. Third, given the higher skillet temperature, the sides of the burger exceed the Malliard temperature, but not enough to burn the patty. Also, the non-centered distribution of the final internal temperature distribution is as expected due to the higher temperature of the second side at the start of the cooling phase.

[1] www.cnet.com/news/size-your-burgers-your-way/

[2] www.livestrong.com/article/545220-how-to-cook-a-hamburger-on-a-griddle/

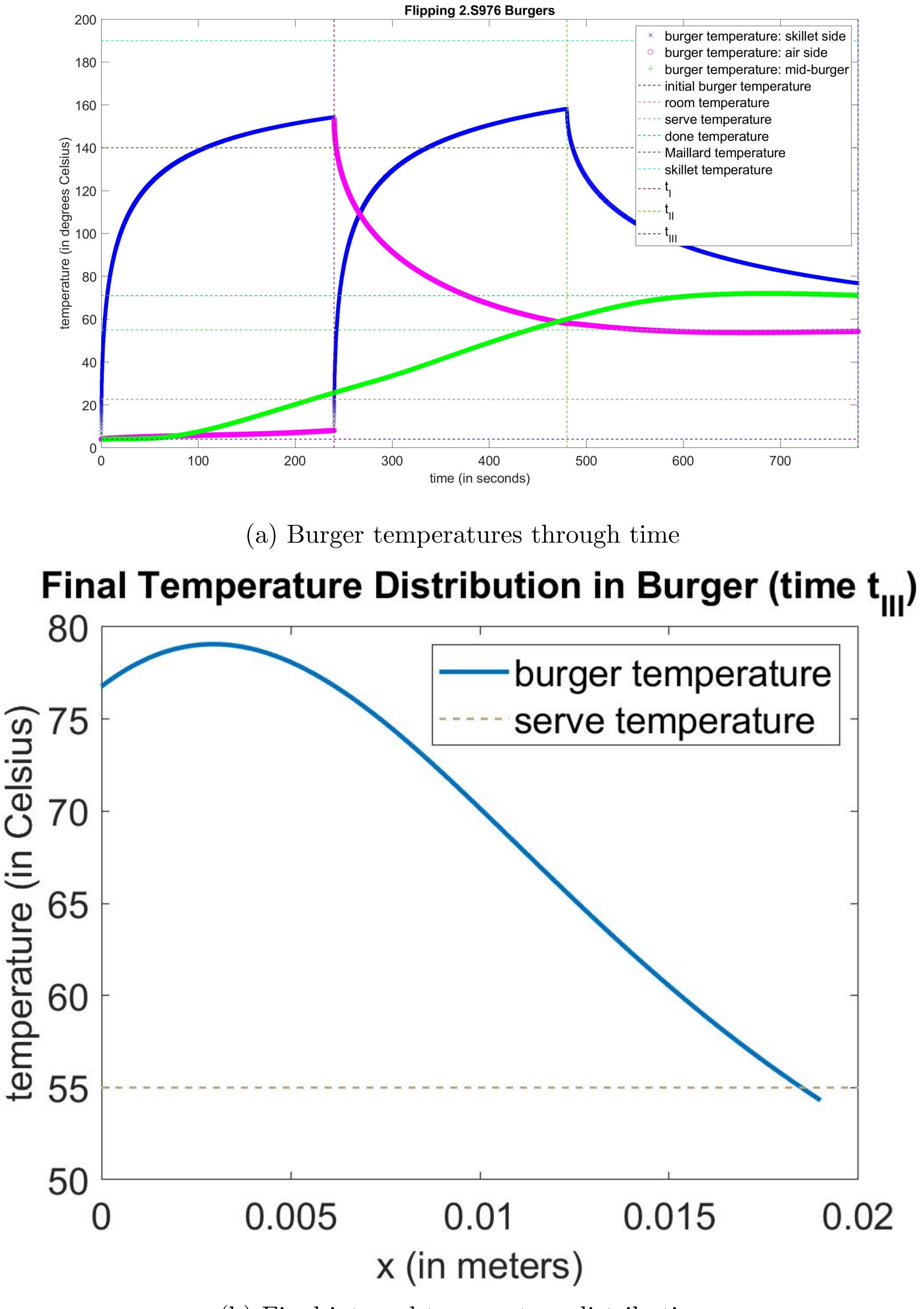
[3] www.cooksillustrated.com/how_tos/9578-why-burgers-need-a-rest



(a) $[p = 1, \theta = 1]$

(b) $[p = 2, \theta = \frac{1}{2}]$

Figure 15: Convergence plot of burger model using two different numerical approaches.



(a) Burger temperatures through time

(b) Final internal temperature distribution

Figure 16: Temperature plots of the burger model using instructions from online sources.

# 4 The FE-FE Method for 4th-Order BVPs (Bending): Xylophone

The finite element method can be extended to understand the eigenvalues of a system, in a physical case, the resonant frequency of a structure. Resonant frequencies can be identified from the governing physical equations of a system, in this case, a beam, through a modal representation. Knowledge of the boundary and initial conditions is required to derive this representation, but afterwards it can be rearranged into a clear eigenproblem. Three types of end conditions that yield different combinations of essential and natural boundary conditions are: clamped, free, and simply supported. Further substitutions yield a form that is similar to the governing differential equations and homogenous boundary conditions from previous chapters. Two types of basis functions are used for this method: one that is either zero or one and a zero derivative at every node, and another that is zero at every node, but the derivative is either zero or one. These basis functions are called Hermitian ($C^1$) basis function, and they accounts for the two degrees of freedom at every node.

In the beam bending case, the modal representation takes the form:

$$u(x,t) = \sum_{k=1}^{\infty}(c_1^{(k)} \cos\omega_n^{(k)} t + c_2^{(k)} \cos\omega_n^{(k)}) u^{(k)}(x) \tag{47}$$

The tuning of a xylophone bar utilizes the FE-FE eigenproblem to calculate the physical parameters and resulting resonant behavior based on target input parameters. The dimensional form of the governing eigenproblem is derived from fundamental beam bending equations specified by the geometric constraints. The FE treatment of these equations in matrix representation is derived in the final equation. The quartic cut-out in the beam is used to change the resonant behavior of the structure, and additionally holes are placed at the zeros of the fundamental node to minimize the force on the bar - mimicing a free-free supported beam. The fundamental frequency corresponds to the third eigenvalue of the beam, and the fourth, fifth corresponds to the first, second harmonic, and so on. The objective of the xylophone problem is, given a desired fundamental frequency, to tune that fundamental and first harmonic to a target ratio - either "quint" tuning where the target ratio is 3, or "double-octave" tuning, where the ratio is 4. This ratio is what gives the produced sound a desireable timbre. Based on the target fundamental frequency, an associated length scaling is computed which satisfies the target behavior.

$$\frac{d^2}{dx_d^2}\left(\frac{E_d W_d H_d^3(x_d)}{12}\frac{d^2 u_d^{(k)}}{dx_d^2}\right) = \lambda_d^{(k)} \rho_d W_d H_d(x_d) u_d^{(k)} \tag{48}$$

$$0 < x_d < L_d \tag{49}$$

$$u_{dxx}^{(k)} = u_{dxx}^{(k)} = 0 = u_{dxx}^{(k)}(L_d) = u_{dxxx}^{(k)}(L_d) \tag{50}$$

$$M_d(0) = V_d(0) = 0 = M_d(L_d) = V_d(L_d) \tag{51}$$

$$\underline{A}\underline{u}_h^{(k)0} = \lambda_h^{(k)} \underline{M}^{inertia} \underline{u}_h^{(k)0} \quad \text{for} \quad k = 1, 2, ..., n \tag{52}$$

The location of holes is deliberate in the geometry of a xylophone bar: they are located at the nodes of the fundamental frequency, which allows the sound to resonate properly. In order to find this location in a Matlab model, a simple algorithm is used. It begins by cycling through every element and finding the ones where the coefficient-modified displacement basis function have values of different signs at each boundary node. This means that the line must cross zero within the element given that the element size is sufficiently small. Next, the fzero Matlab function is used to find the actual zero location within the element based on the finite element approximation. Finally the local position is translated into the global position based on the left node location of the element it was found in.

```
x_hat_hole = [];
xhole_d = [];
for m=1:n_el
    if u3(lg2(1,m))*u3(lg2(3,m))<0
        find_zero = @(x) ones(1,4)*(u3(lg2(1:4,m)).*hshape_fcn(x,h(m)));
        x_hat_hole = fzero(find_zero,[0:1]);
        xhole_d =  [xhole_d (xpts(lg(1,m))+h(m).*x_hat_hole).*L_d];
    end
end
```

Parameters and results from a vibrating bar experiment conducted to explore Chlandi patterns is used to validate the proper functionality of the Matlab code. Recorded physical properties along with the theoretical fundamental frequency were used as inputs and the outputs of the fundamental and first harmonic along with the length of the beam were compared to determine accuracy. Additionally, the p2 interval parameter had to be changed to $[1.0, 1.0]$ because the beam used in the experiment does not have a curved cutout like a xylophone bar, so the parameter change prevents the program from implementing that geometry.

$$L = 1.275m \quad A = h \times b = 0.01 \times 0.075m \tag{53}$$

$$\rho = 7800Kg - m^{-3} \quad E = 2.1 \times 10^11Nm^{-1} \tag{54}$$

The program calculated the fundamental and first harmonic as 32.8 Hz and 90.415 Hz, respectively, and a beam length of 1.275 m. This agrees with the theoretical results and the beam geometry as shown in figure 17. Although the fundamental and beam length values matched the theoretical results and beam data exactly, the first harmonic had an error of $-0.028\%$, given its theoretical value of 90.44.

The case of tuning a xylophone bar demonstrates the full functionality of the provided eigenproblem program `xylo_bar_design3`. The parameters used to test were as follows:

```
frequency3target_d = 349.23; %F4 - the target fundamental frequency
R_target = 3; %"quint" tuning - the target frequency ratio
Hmax_d = 0.015; % the maximum height of the bar
xstar = 0.05; % the nondimensional location of the start of the modified
% section of the bar
p2_interval = [.05,1.0]; %the design variable p2 is permitted to vary in the interval
Ebar_d = 1.4e10; % Young's modulus
rhobar_d = 835; % density
justcalc_L_d = false;
suppress = true;
```

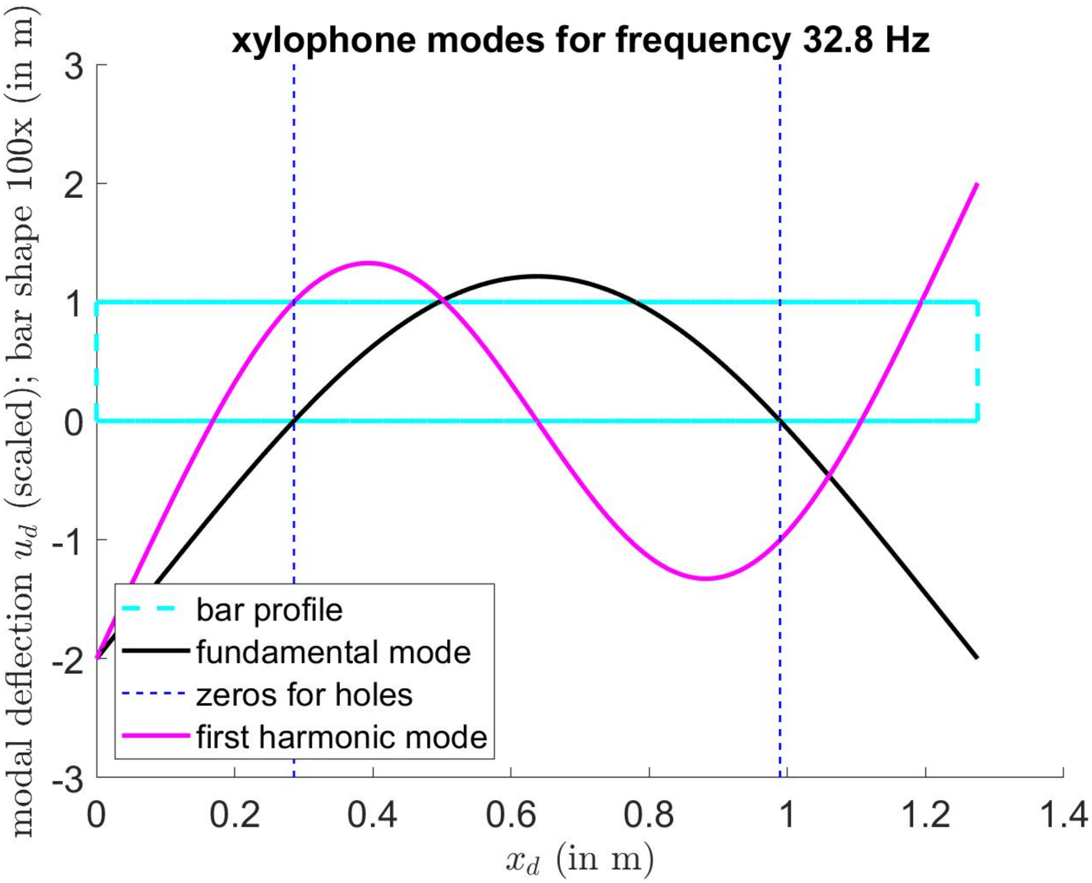


Figure 17: Beam resonance modes based on the Carestra case.

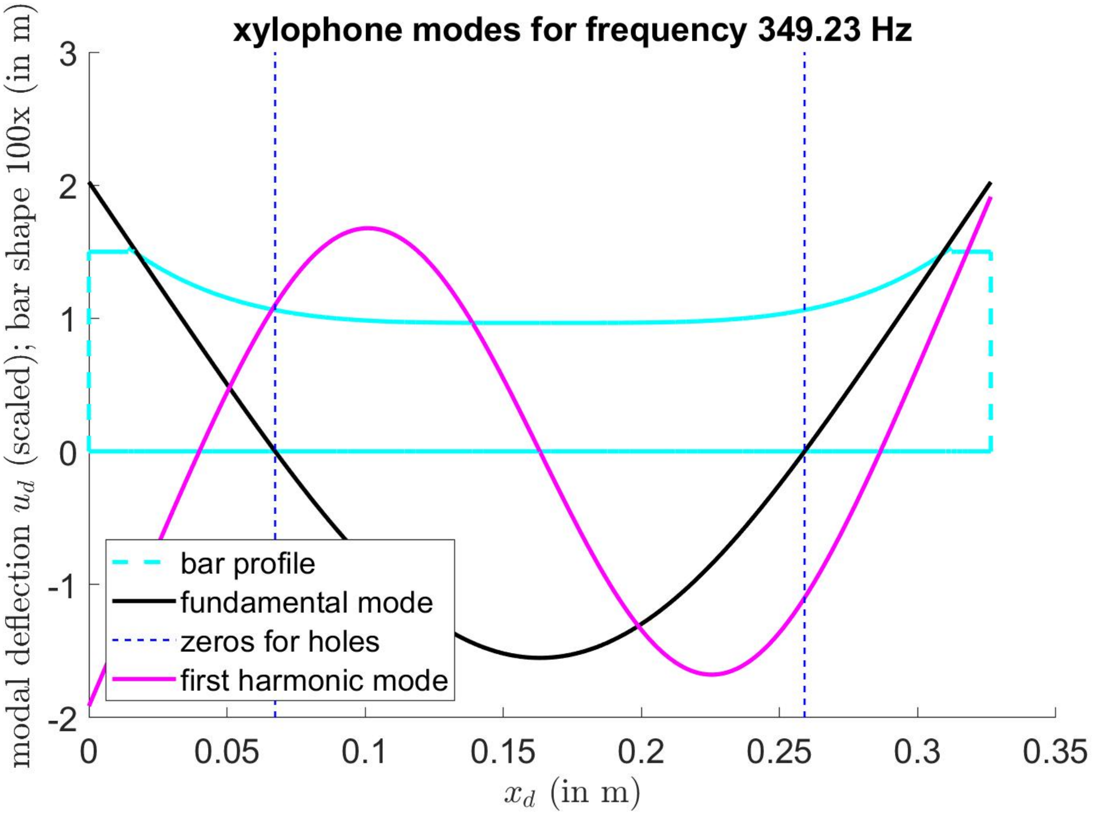


Figure 18: Fundamental and first harmonic xylophone frequencies with optimal shape profile.

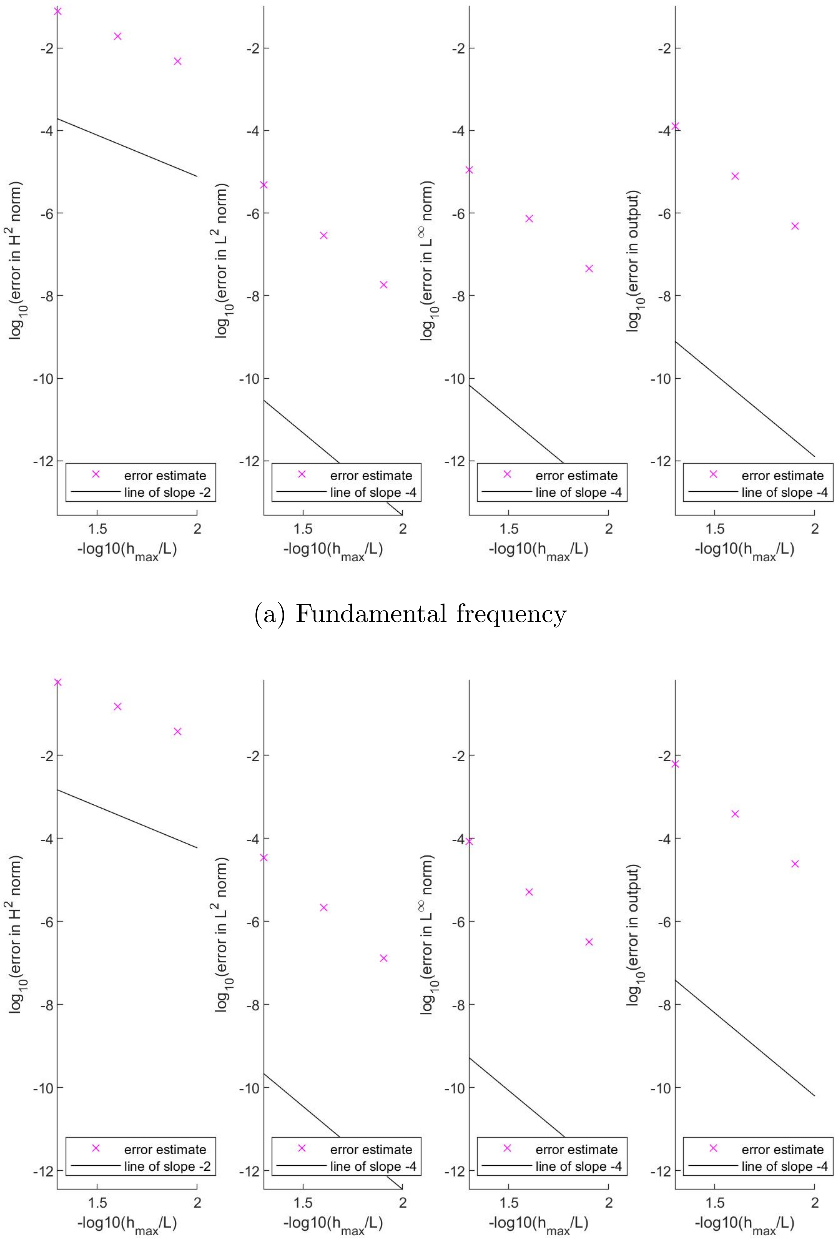
Error and convergence analysis provides a metric for the performance and functionality of the program. The values of interest are the fundamental and first harmonic frequencies and the ratio of the first harmonic over the fundamental. The error for the ratio is not provided as an output, so it can be easily derived by taking the maximum ratio of the errors expressed as a fraction below. The following results were achieved along with figure 18. Note that the figure also shows evidence of proper implementation of the modified function as seen by the line intersecting the fundamental node:

$$\text{Fundamental Freq:} f_d^3 = 349.23 \pm 5.857 \times 10^{-6}\text{Hz}$$

$$\text{First Harmonic Freq:} f_d^4 = 1044.6 \pm 1.958 \times 10^{-6}\text{Hz}$$

$$\text{Ratio:}\frac{f_d^4}{f_d^3} = 2.9913 \pm 0.3343 \quad P_2^{opt} = 0.6438 \quad L_d = 0.3264\text{m}$$

The outputs of the initial run of `xylo_bar_design3`, the shape design parameter p2opt can be used as an input in the p2 interval variable to observe the error convergence behavior of the function. The error estimators are reliable because they appear to converge at the correct rate. For the same mesh size, the error for the fundamental frequency is expected to be lower than the error for the first harmonic because the the latter is a higher order curve that is harder to approximate, and the plots in Figure 19 agree with this assessment.



(a) Fundamental frequency

(b) First harmonic frequency

Figure 19: Error estimates for xylophone bar tuning showing proper convergence.

Based on the hypothesis that the human ear cannot differentiate pitches that differ less than 10 Hz, the h-value chosen for this function is too small because the error range for both frequencies are several orders of magnitude smaller than 10 Hz. Even without any refinements the errors would still be small enough to be indistinguishable from the theoretical exact value. Based on the assumption that the bar is treated like an Euler-Bernoulli beam, it is predicted that the functions will be more accurate for bars tuned to low frequencies compared to high frequencies because of the lower curvature of the waveform. Given a constant height of the bar, the decrease in aspect ratio as the bar gets smaller for tuning higher frequencies would make it more difficult to approximate the solution. Smaller h-values would be required to achieve the same degree of accuracy for tuning to higher frequencies. As evidence, the results of tuning a bar to a higher frequency (698.46 Hz = F5) is given below:

$$\text{Fundamental Freq:} f_d^3 = 698.46 \pm 1.171 \times 10^{-5}\text{Hz}$$

$$\text{First Harmonic Freq:} f_d^4 = 2089.3 \pm 3.916 \times 10^{-6}\text{Hz}$$

$$\text{Ratio:}\frac{f_d^4}{f_d^3} = 2.9913 \pm 0.3343 \quad P_2^{opt} = 0.6438 \quad L_d = 0.2308\text{m}$$

This statement, however hinges on the assumption that the h-value is independent on the length. If h decreased with h for the same number of elements, then we expect the FE method to produce relatively the same errors. Alternatively, $\tau_d$ is a factor that occurs in the denominator of the error estimates and is directly correlated to the length. Therefore, as the length decreases, the error is expected to increase not because of h, but because of $\tau_d$.

To more accurately represent the actual behavior of the xylophone bar system, the function can be modified to account for the support strings that are threaded through each of the two holes in the bar. A simplified version of this condition can be represented by a beam with a lumped spring on one end, where the spring force is contributing to the shear force on the right end. A modified energy functional that incorporates this "Robin" boundary condition yields a new stiffness matrix in the form:

$$A_{ij} = \int_0^L EI \frac{d^2\varphi_i}{dx^2}\frac{d^2\varphi_j}{dx^2} + k_s\varphi_i(L)\varphi_j(L), \quad 1 \leq i,j \leq 2 \cdot n_{node} \tag{55}$$

Similar to the modification to the A matrix in chapter 2, additional lines can be added to the function `impose_boundary_cond`to accomplish this. The matrix elements that are affected by this modification are only those that correspond to the second to last phi function before refinement because it is the only basis function that is nonzero at that node. Therefore, a conditional statement can be written to implement this addition only at those specific indexes assuming that the new parameter `f_Omega_fcn` has already been initialized in `xylo_bar_design` as a vector of with all values k (the spring constant). The following line would be inserted after line 56 in the function because the spring force does not affect the displacement of the beam and is considered to be a "Robin" condition.

```
% xylo_bar_design.m line 188
probdef.f_Omega_fcn = @(x) k.*ones(length(x),1);
```

```
%impose_boundary_cond.m line 57
f_Omega_fcn = probdef.f_Omega_fcn(0);
A(2*(n_el0+1)-1,2*(n_el0+1)-1) = A(2*(n_el0+1)-1,2*(n_el0+1)-1) + f_Omega_fcn;
```

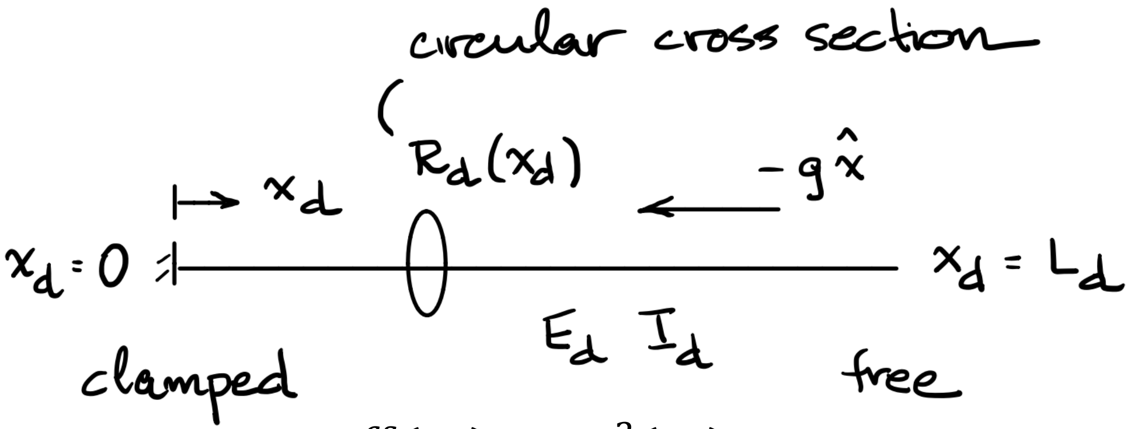
# The FE Method for 1D 4th-Order BVPs (Bending): Self-Buckling

Patrick Shin

2.S976 Finite Element Methods for Mechanical Engineers

May 16, 2019

# Self-buckling



$$A_d^{cs}(x_d) = \pi R_d^2(x_d)$$

$$I_d(x_d) = \frac{\pi}{4} R_d^4(x_d)$$

Volume: $V_d = \pi R_d^2 L_d$

Impose axial load and moment balance to derive:

$$\frac{d^2}{dx^2}\left(R^4 \frac{d^2u}{dx^2}\right) = \lambda\left(-\frac{d}{dx}\left(P\frac{du}{dx}\right)\right) \qquad 0 < x < 1$$

$$u = u_x = 0 \qquad \text{at } x = 0$$

$$u_{xx} = (R^4 u_{xx})_x = 0 \qquad \text{at } x = 1$$

Where $\lambda^{(1)} = \gamma_c$ -> no self-buckling for $\gamma < \gamma_c$

# FE Methods for Self-Buckling

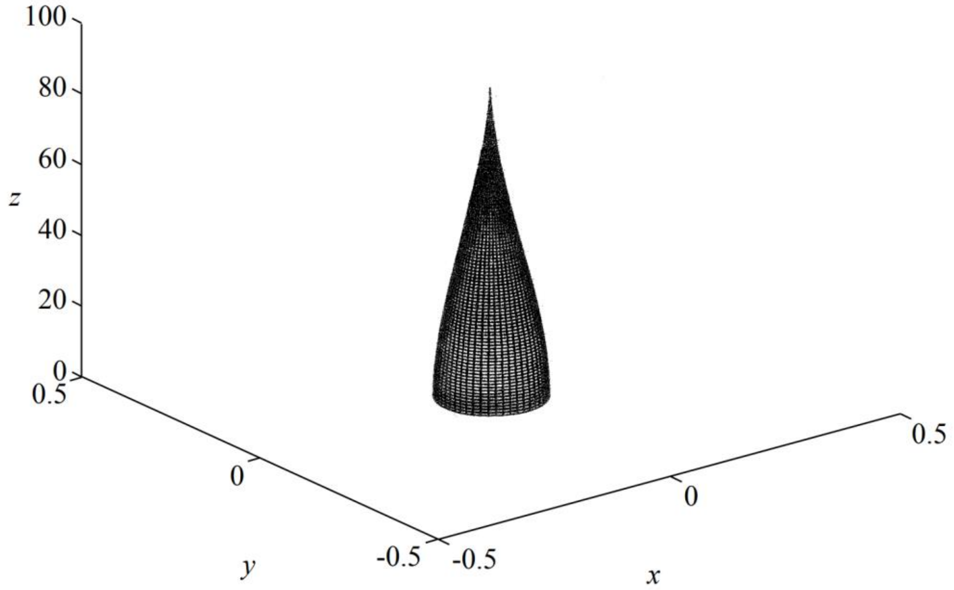
- Matrix Form
  - $Au_h^0 = \lambda_h K^{ax} u_h^0$
  - Where $\lambda_h = \gamma_{ch}$
  - Remove row and col 1 and 2 from A and K
- $A_{ij} = A_{ij}^N = \int_0^1 R^4(x) \frac{d^2\varphi_i}{dx^\wedge 2} \frac{d^2\varphi_j}{dx^2} dx$
- $K_{ij}^{ax} = K_{ij}^{axN} = \int_0^1 P(x) \frac{d^2\varphi_i}{dx^\wedge 2} \frac{d^2\varphi_j}{dx^2} dx$
- Error
  - A Priori: $0 < \lambda_h - \lambda \sim C_u \underbrace{\left(\frac{h}{h_0}\right)^{\overset{r=4}{4}}}_{2^{-4\ell}} \text{ as } h \to 0 \qquad = \left(\frac{1}{16}\right)^{\ell}$
  - A Posteriori: $\log_2 C_u - 4\ell \text{ as } h \to 0$

# Optimization Problem

- Fixed Volume: $\int_0^1 G(x)dx = 0$
- Minimum Radius (R_min): $G(x) \geq -1 + R_{min}^2 \equiv G_{min}(\succ -1)$
- Continuous, weakly differentiable: $|G'(x)| \leq S_{max}$
- Objective
  - Maximize $\gamma_c$ over P1 : $\gamma_c^{opt}$
  - Choose $L_d^{opt}$ = $\left(\frac{\gamma_c^{opt}\epsilon_d V_d}{4\,\pi\rho g}\right)^{\frac{1}{4}}$
- Figure of Merit: $\frac{L_d^{opt}}{L_{d,cyl}^{opt}}$ (at fixed volume) = $\left(\frac{\gamma_c^{opt}}{\gamma_{c,cyl}}\right)^{\frac{1}{4}}$

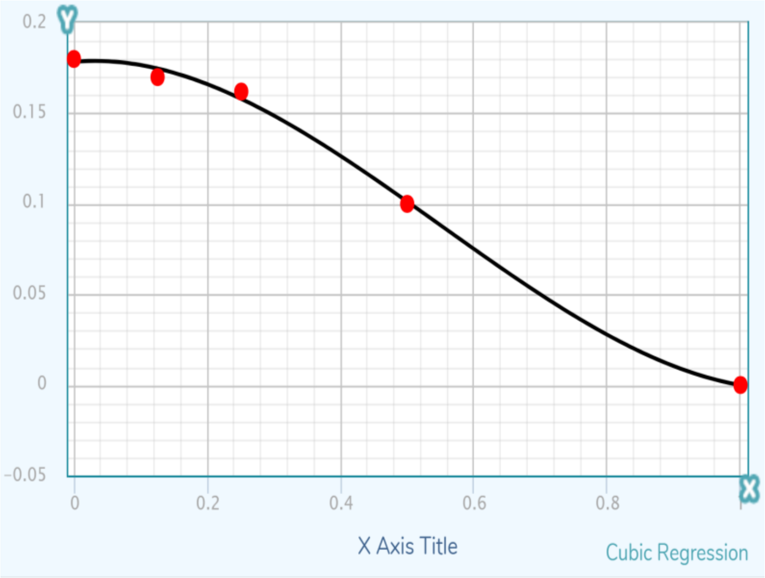
# Shape Family

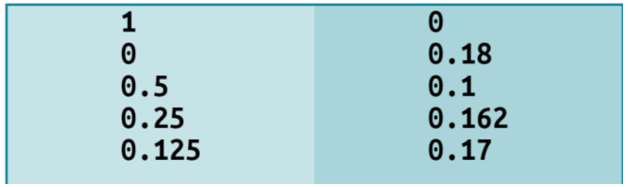
- Cubic regression- third order polynomial
- Roughly based on 2.080 "tallest column" plot – pixel measurements
- Used wolfram alpha to determine +C to satisfy CV
- Scaled by 10 so G(1) = -1 (close to R_min)



https://ocw.mit.edu/courses/mechanical-engineering/2-080j-structural-mechanics-fall-2013/course-notes/MIT2_080JF13_Lecture10.pdf

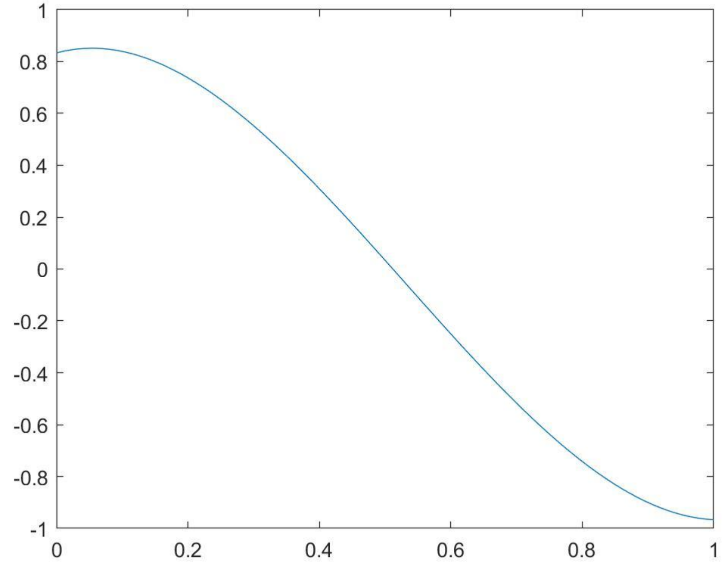
# Shape Family





| | |
|---|---|
| 1 | 0 |
| 0 | 0.18 |
| 0.5 | 0.1 |
| 0.25 | 0.162 |
| 0.125 | 0.17 |

Points derived from pixel measurements
https://mycurvefit.com/



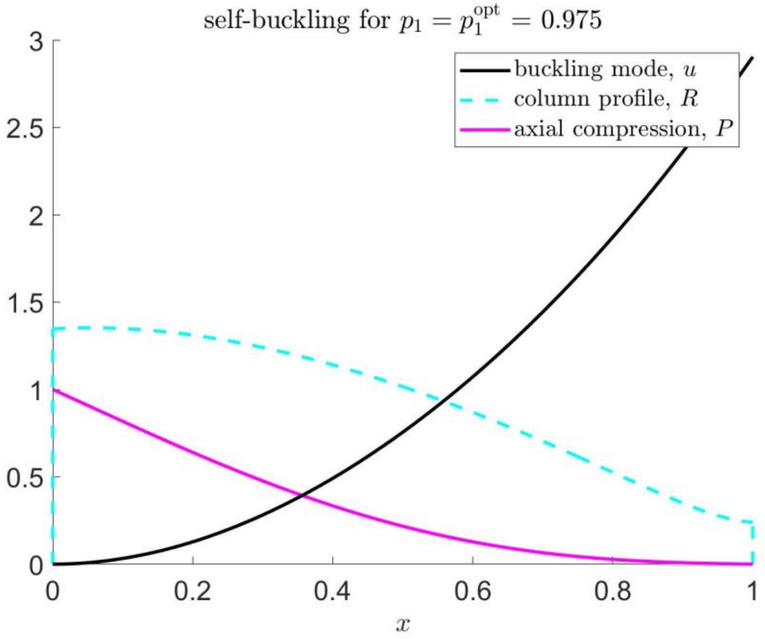
Test plot used to confirm correct shape after constraining and scaling

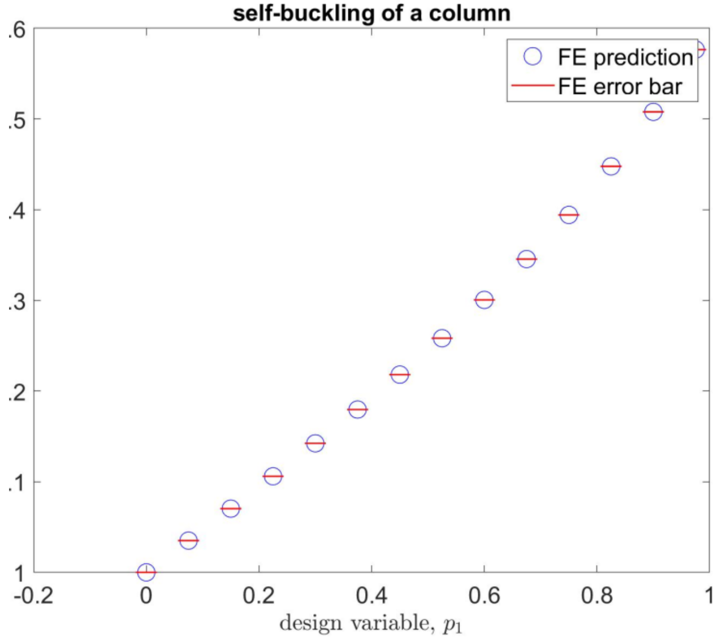
# Gfunc_sver

- G =  ampl1.*10.*((1./12) + 0.068.*x - 0.664.*x.^2 +0.416.*x.^3);
- P1 used as amplitude scaling in order to determine accuracy of initial guess
- P2 considered as factor for coefficient terms, but complicated satisfying constraints
- Scan range: [0,1.5]
- 4 Refinements

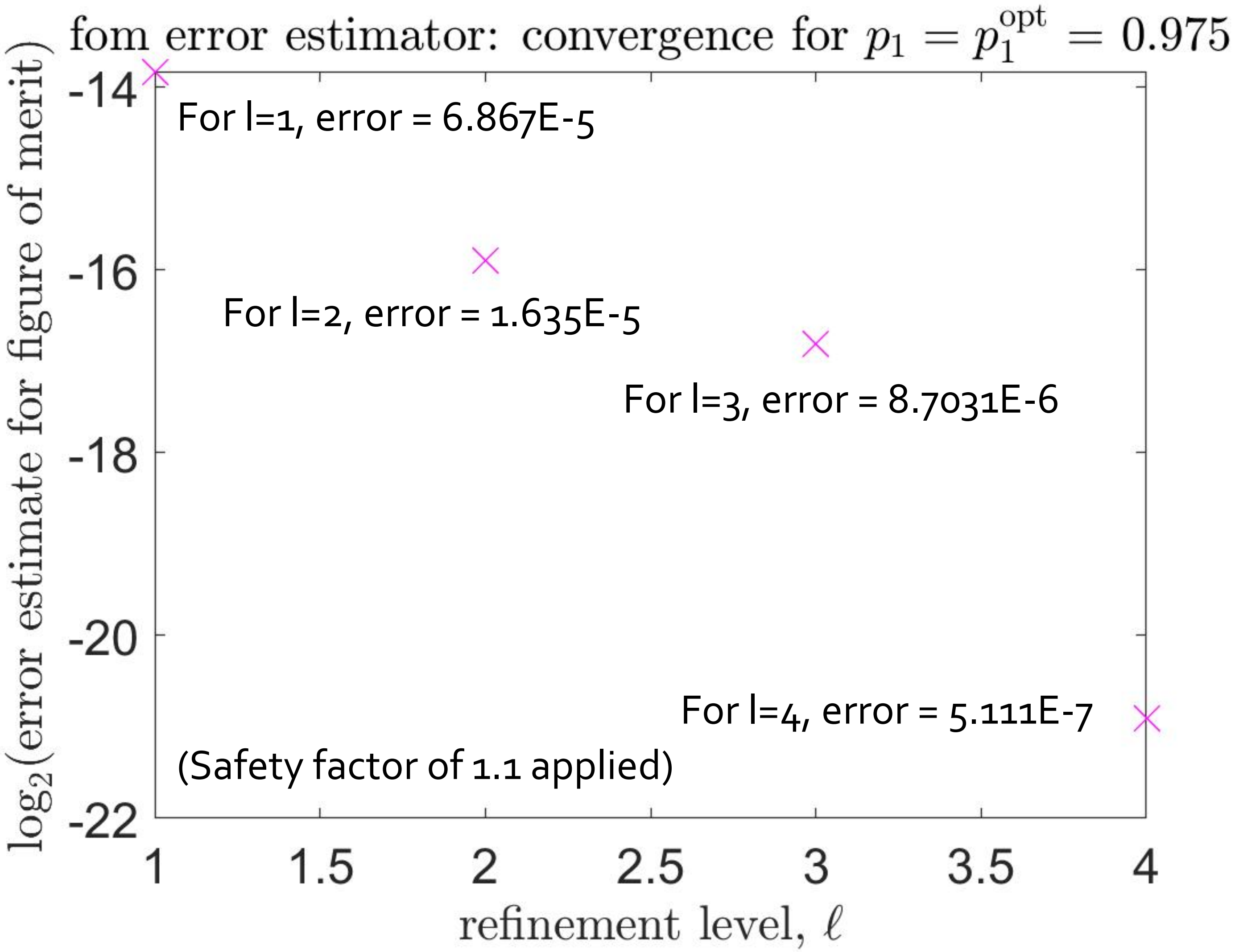
# Results

- Max FOM: 1.577
- P1_opt = 0.975, confirms initial prediction of optimal solution
- P1_opt < 1 due to R_min constraint





# Error Results



For l=1, error = 6.867E-5

For l=2, error = 1.635E-5

For l=3, error = 8.7031E-6

For l=4, error = 5.111E-7

(Safety factor of 1.1 applied)