MECH 309—Winter 2022—Final Project Report—April 8, 2022

Abdul Ashour 260863398, Kuvish Bussawah 260683262, Paul Hinta 260850630, Justin Hirsch 260862841

1 Errors and stability in numerical methods

An error is defined as the difference between a true value and an approximate, or an estimate, of that same value. The most common ones we interact with are the round-off error, truncation error and discretization error. Round-off error is when a numerical value between a said number is against its closest real number value. An example is where we would round up the number 3.9 to 4. This attempt identifies what the rounding distance is when we discuss algorithms and we could also call this quantization error. A truncation error occurs when an approximation is involved in numerical analysis, where the error factor is related to how much the approximate value varies from the actual value itself. Discretization involves converting or partitioning variables or continuous attributes to nominal attributes, intervals and variables. As a type of truncation error, the discretization error focuses on how much a discrete math problem is not consistent with a continuous math problem.

However, errors can also be of use for statistics, computer programming, advanced mathematics and much more. Evaluating errors provides critical information of the accuracy of the result, especially when chance and probability are being analyzed.

The notion of stability is often used to see how much an error can have an effect, or how bounded it can be. If an error stays at one point in an algorithm and doesn't aggregate further as the calculation continues, then it's considered a numerically stable error. This happens when the error causes only a very small variation in the formula result. If the opposite occurs and the error continues to propagate as the calculation continues, then it is considered numerically unstable. Furthermore, stability is also of use when we want to see output data and input data can affect each other.

For a forward error, the variance created when comparing the two values, is when the output result is different from its supposed true output value. The backward error would be the opposite way around. The variance occurring between the approximated input value compared to the real input value directly impacts the output data. In most scenarios, error analysis, either relative or absolute, helps to estimate the forward error, but sometimes it is much simpler to instead estimate the backward error. Intuitively speaking, a condition number is a measure of the change in the solution due to a change in the data. If the condition number is large, where it has a value higher than 1, it means that a small change in the input has a large impact on the solution. This would be called an ill-conditioned or sensitive scenario. If the condition number is small, a large change in input has a small impact on the solution, it would be called a well-conditioned or sensitive scenario. Finding the condition of a problem is an important step in error analysis, however it does not provide sufficient information to know if the process from input to output will yield an accurate solution. A stable algorithm is one where when applied to a given problem with an exact input data gives an exact solution even if the input data was to be changed to an approximate input data, it would give out the same said exact solution.

2 Inverted Flags

2.1 Plates & Beams–Mechanical System

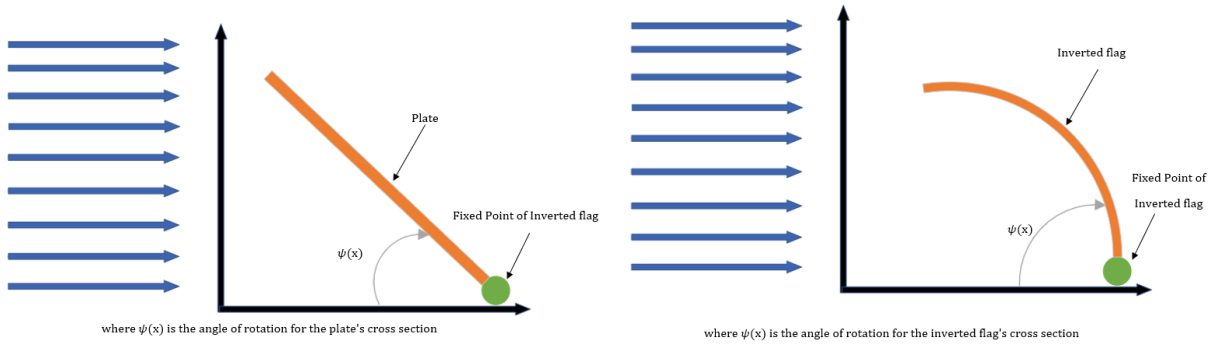

Figure 1: Behavior of plate and beam with straight cross section

2.2 Challenges with $\psi(x)$

Obtaining an expression for $\psi(x)$ is difficult for the following reasons: $\psi(x)$ is governed by a non-standard differential equations (DE); $\psi(x)$ is a compound function involving nested sinusoidal functions; In cases where 'special' mathematical constants (1, 0, π, i, e) aren't relevant, mixing sinusoidal & hyperbolic functions becomes very messy to solve analytically.

2.3 Approximation of Governing Transcendental Equation

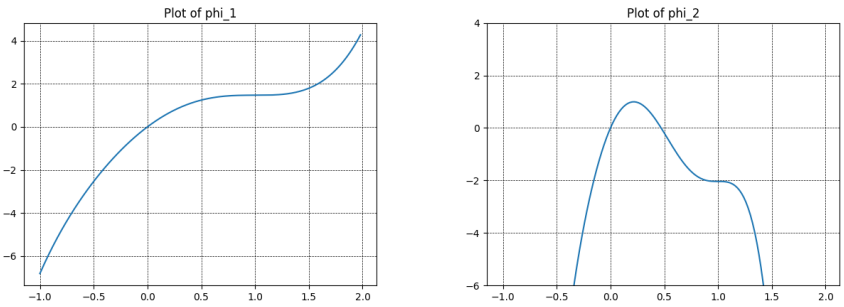

Figure 2: Plot of $\phi_1(x)$ & $\phi_2(x)$

2.4 Galerkin's Procedure

DEs are difficult to analyze numerically; Galerkin's method discretizes the DE in space. To do this, we multiply the DE by a weight function $w(x)$ and form the intergal over the region of interest; in some cases, the order of the highest term can be reduced by integrating by parts. Next, we choose the order of interpolation (linear, quadratic, etc.) & the associated shape functions, N_i. The approximation is $f \approx f(x) = \sum_{i=1}^{m} N_i(x) f_i$. The integrals are evaluated over each element, either exactly or numerically, to set up a system of equations in the unknowns f_i, which is solved at the end.

2.5 Analysis & plot of $f(q_1)$

Next, the goal is to plot $f(q_1)$ as applied by Galerkin's procedure. To do this, we can analyze function given in section 2.4 in the following series of steps:

1. Discretize x on $[0, 1]$ and solve the linear component (the single integral component) by applying the trapezoid rule

2. The nonlinear component (the double integral) can easily be solved if the values of x and q_1 are given. Discretize x on $[0, 1]$ and q_1 on $[-5, 5]$ and apply the trapezoid rule on the inside integral (the integral in ds from x to L).

3. This will give us the numerical value of a function $I(x, q_1)$ (I for integral); the outside integral can now be solved by applying the trapezoid rule in dx.

4. We will now have a series a functions $f(q_1)$ that are numerically represented for a discretized set $q_1 \in [-5, 5]$. For each point q_1, we can plot $f(q_1)$.

Below is a visualization that can help us understand how discretizing in x and q_1 helps us find $f(q_1)$. Note that each element in each matrix is actually a numerical entry, thus it is only possible to plot the function numerically.

$$\int_x^L \begin{bmatrix} \langle x=0, q=-5\rangle & \dots & \langle x=1, q=-5\rangle \\ \langle x=0, q=-4\rangle & \dots & \langle x=1, q=-4\rangle \\ \dots & & \dots \\ \langle x=0, q=5\rangle & \dots & \langle x=1, q=5\rangle \end{bmatrix} ds \rightarrow \int_0^1 \begin{bmatrix} I(x=0, q=-5) & \dots & I(x=1, q=-5) \\ I(x=0, q=-4) & \dots & I(x=1, q=-4) \\ \dots & & \dots \\ I(x=0, q=5) & \dots & I(x=1, q=5) \end{bmatrix} dx \rightarrow \begin{bmatrix} f(q_1=-5) \\ f(q_1=-4) \\ \dots \\ f(q_1=5) \end{bmatrix}$$

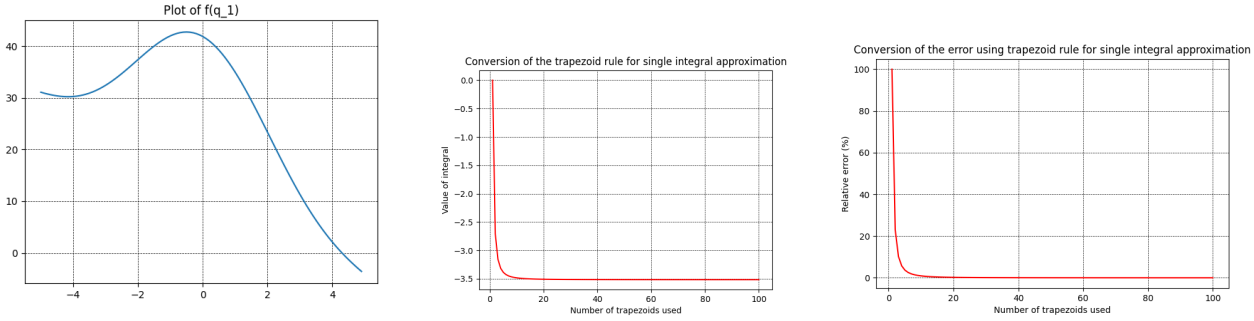

Figure 3: The plot of $f(q_1)$

The above plots and the respective error were used to evaluate the rate of convergence of the implementation of a single integral towards an accurate solution using 1000 data points. Inspection of figure 3 reveals that the integral has a quadratic rate of convergence for both solution and error. A similar implementation is applied when solving a double integral; however, the error is squared.

2.6 Fixed-Point Method

In a fixed-point scheme, we want to represent find a representation of a function $f(q_1)$ such that:

$$f(q_1) = g(q_1) - q_1 \implies g(q_1) = q_1$$

Conveniently, from section 2.5, we know that we can express $f(q_1)$ as:

$$f(q_1) = D\int_0^1 \phi_1(x)\psi_h''(x)dx + F_N \int_0^1 \int_x^L \phi_1(x) cos(\psi_h(x) - \phi_h(s))dsdx$$

From our numerical analysis, we know that the term $D\int_0^1 \phi_1(x)\psi_h''(x)dx$ is a linear function of q_1, approximately equal to $-3.516q_1$, which is convenient for fixed point analysis. We can call this the $g(q_1)$ function and say:

$$F_N \int_0^1 \int_x^L \phi_1(x) cos(\psi_h(x) - \phi_h(s))dsdx = -D\int_0^1 \phi_1(x)\psi_h''(x)dx \approx 3.516q_1$$
$$\implies \frac{F_N}{3.516}\int_0^1 \int_x^L \phi_1(x)\cos(\psi_h(x) - \phi_h(s))dsdx = q_1$$

This gives us a fixed-point scheme that allows us to solve for q_1, as shown in figure 4 below.

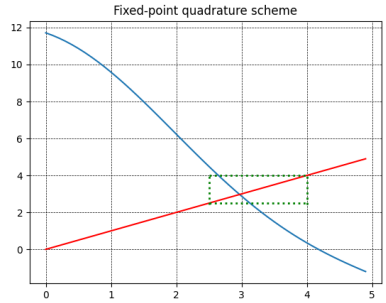

Figure 4: The fixed-point scheme to solve $f(q_1)$

By inspection the domain $[a, b]$ for which $g([a, b]) \subset [a, b]$, i.e. $2.5 \leq q_1 \leq 5$, we cannot confirm that the absolute value of slope of $g(q_1)$ is less than 1, i.e. we cannot say for sure that $0 < |g'(q_1)| < 1$. Thus by Theorem 2.4, we cannot confirm that the fixed-point iteration converges. It is likely better to use a different method to compute q_1.

2.7 Numerical approximation of q_1 in 1D

As mentioned in the previous section, we will use a different method to approximate q_1. As mentioned in section 2.5, the actual mathematical representation of $f(q_1)$ is not known; it therefore isn't practical to compute the derivative, so Newton's Method should not be used. The next closest option is the Secant method, which is very convenient to use numerically. Although we don't have the exact expression of $f(q_1)$, we assume that our approximation is good enough and therefore the typical process that we have seen in chapter 2 is sufficient.

Doing this, and with starting guesses $p_{1,0} = 4$, $p_{1,1} = 4.1$, and $\epsilon = 0.001$ gives $q_1 = 4.278$ (to three digits). Note that this is approximately equal to the fixed point shown in figure 4. Inspection of figure 4 reveals that the rate of convergence is quadratic. Note that all initial guesses used in this plot were 0 and 0.1; however, using a guess closer to the initial root reduces the number of steps. This is evident in section 2.7, where the initial guess only required three steps to converge to three digits of accuracy.

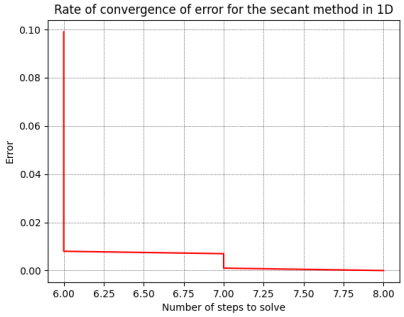

Figure 5: Error analysis of our implementation of the Secant Method in 1D

2.8 Transverse deflection of the beam using q_1

$$w(x) = \int_0^x \sin(\psi(s))ds \approx \int_0^x \sin(q_1\phi_1(s))ds$$

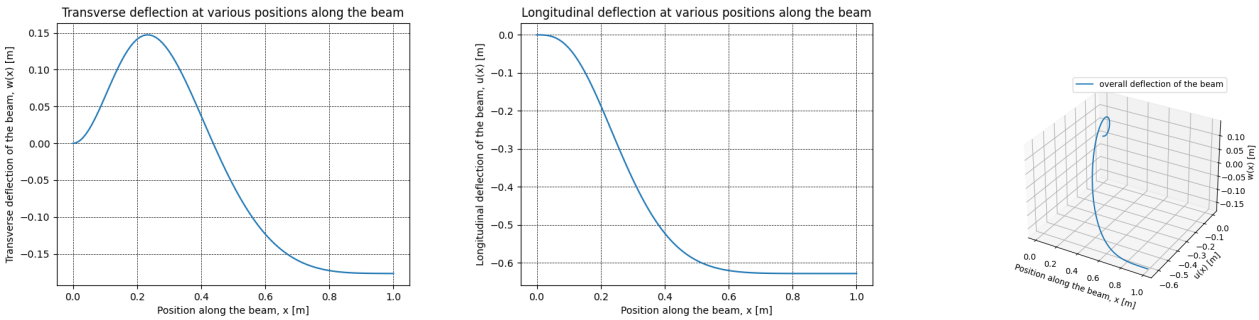

Figure 6: Transverse, longitudinal, and overall deflection of the beam using one q_i term

2.9 Transverse deflection of beam at different F_N using q_1

Since we're looking only at $\psi(L) \approx q_1\phi(L)$, we will redo the steps above, varying F_N from 0 to 100 in order to find q_1 at different F_N. These values are gathered and plotted in figure 7.

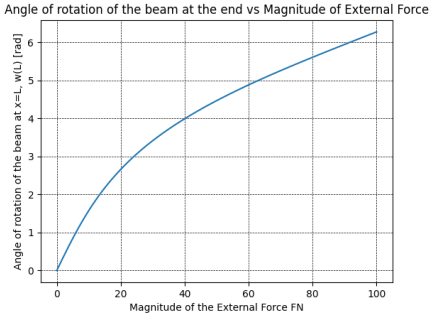

Figure 7: Angle of Rotation $\psi(x)$ of the beam at $x = L$, as it relates to F_N, using one q_i term

2.10 2D Expansion of f—Plots

We now expand the definition of $\psi(x)$ to take into account multiple variables q_i according to equations 1 & 2:

$$\psi \approx \psi_h = \sum_{i=1}^{n} q_i \phi_i(x) \tag{1}$$

$$f_i(q1, ..., q_n) = D\int_0^1 \phi_i(x)\psi_h''(x)dx + F_N \int_0^1 \int_x^L \phi_i(x) cos(\psi_h(x) - \phi_h(s))dsdx \tag{2}$$

We will first explore the system of two variables, q_1 and q_2 (i.e. the 2D case of equation 2.

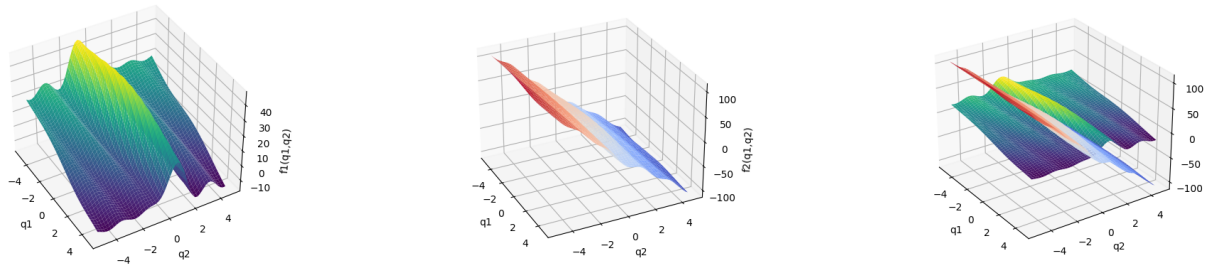

Figure 8: 3D Plots of $f_{1,2}(q_1, q_2)$ and $f_2(q_1, q_2)$

2.11 Fixed-Point Formulation in q_1, q_2

The system of equations $\overrightarrow{f} = \langle f_1(q_1, q_2), f_2(q_1, q_2)\rangle$ includes a linear component, a non-linear component, and a constant component .

$$\overrightarrow{f} = \begin{cases} c_1 q_1 + c_2 q_2 + \sum_0^n C_n \cos\left(a_i q_1 + b_i q_2\right) + C \\ \underbrace{k_1 q_1 + k_2 q_2}_{\text{Linear}} + \underbrace{\sum_0^n K_n \cos\left(d_i q_1 + e_i q_2\right)}_{\text{Nonlinear}} + \underbrace{K}_{\text{Constant}} \end{cases} \tag{3}$$

For some constants $c_1,\ c_2,\ k_1,\ k_2,\ C_n,\ K_n,\ a_i,\ b_i,\ d_i,\ e_i,\ C,$ and K. The linear component of $\overrightarrow{f}$ can be rewritten as:

$$\begin{aligned} \overrightarrow{f}_{\text{linear}} &= \begin{bmatrix} c_1 q_1 & c_2 q_2 \\ k_1 q_1 & k_2 q_2 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} c_1 & c_2 \\ k_1 & k_2 \end{bmatrix}}_{\boldsymbol{B}} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \boldsymbol{B} \times \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \boldsymbol{B} \times \boldsymbol{q}^T \end{aligned}$$

We take $\overrightarrow{g}(\boldsymbol{q}) = \boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$, which is the two-dimensional equivalent of $g(q_1) = q_1$ and will help us with our fixed-point formulation. We can manipulate equation 3:

$$\begin{aligned} \overrightarrow{f} - \boldsymbol{B}\boldsymbol{q} - \boldsymbol{C} &= -\left(\boldsymbol{B}\boldsymbol{q} + \boldsymbol{C}\right) \\ \implies \overrightarrow{g} = \frac{\overrightarrow{f} - \boldsymbol{B}\boldsymbol{q} - \boldsymbol{C}}{-\det\left(\boldsymbol{B}\right)} &= \boldsymbol{q} + \frac{\boldsymbol{C}}{-\det\left(\boldsymbol{B}\right)} = \boldsymbol{q} + \boldsymbol{C'} \end{aligned}$$

Similarly to the 1-D fixed-point case, there is no way to confirm that $\overrightarrow{g}$ will converge by analyzing its partial derivatives. That is, we cannot say for certain that equation 4 below is satisfied. Another method should be used.

$$\left|\frac{\partial g_i}{\partial x_j}\boldsymbol{x}\right| < \frac{K}{n} \qquad \text{for } \boldsymbol{x} \in D, \qquad \forall (i,j) = 1, ..., n \tag{4}$$

2.12 Numerical Approximation of q_1 & q_2 in 2D

We will apply the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm to solve $\boldsymbol{q}$, which is a modification to Broyden's Method, given by the recursive relationship 5 below. The initial guesses are $\boldsymbol{q_0} = \begin{bmatrix} 4.25 \\ 0.3 \end{bmatrix}$ (using the results for q_1 & q_2 derived by solving in 1D) and $\boldsymbol{A_0} = \boldsymbol{I}$.

$$\boldsymbol{A_{k+1}} \leftarrow \boldsymbol{A_k} - \frac{\boldsymbol{A}_k \Delta \boldsymbol{x}_k \Delta \boldsymbol{x}_k^T \boldsymbol{A}_k}{\Delta \boldsymbol{x}_k^T \boldsymbol{A}_k \Delta \boldsymbol{x}_k} + \frac{\Delta \boldsymbol{f}_k \Delta \boldsymbol{f}_k^T}{\Delta \boldsymbol{x}^T \Delta \boldsymbol{f}_k} \tag{5}$$

Solving the system using the iterative scheme (5), we find $\boldsymbol{q} = [q_1, q_2]^T = [1.759, -0.836]$; note that the value q_1 is vastly different than the one found in section 2.8. Thus, the $\phi_2(x)q_2$ term is critical to the approximation.

2.13 Deflection of the beam at using two q_i terms

Based on figure 9, it appears that the contribution of the $\phi_2(x)q_2$ term is critical to the approximation of $\psi(x)$.

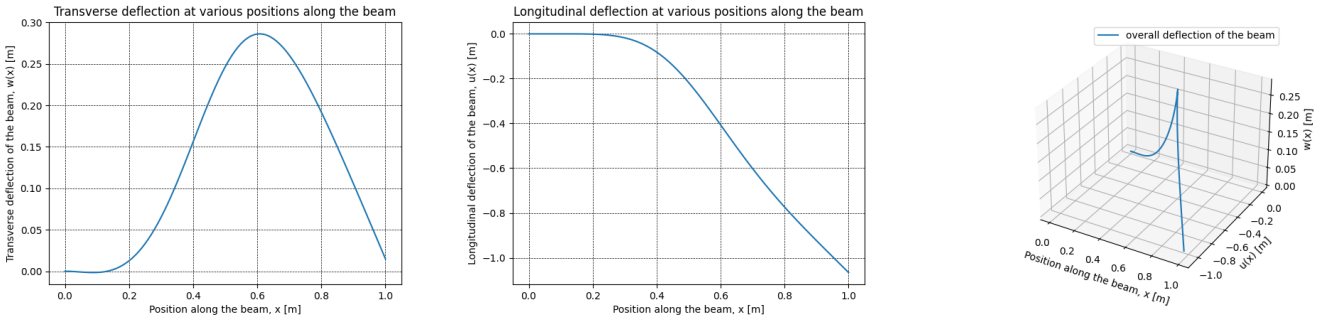

Figure 9: Transverse, longitudinal, and overall deflection of the beam using two q_i terms

2.14 Numerical Approximation of q_1, q_2, & q_3 in 3D

Equations 1 & 2 now use three q_i terms. The process described in section 2.12 is repeated, this time taking $\overrightarrow{f} = \langle f_1(q_1, q_2, q_3), f_2(q_1, q_2, q_3), f_3(q_1, q_2, q_3)\rangle$. This yields $\boldsymbol{q} = [q_1, q_2, q_3]^T = [1.777, -0.848, -0.017]$.

2.15 Deflection of beam at different using three q_i terms

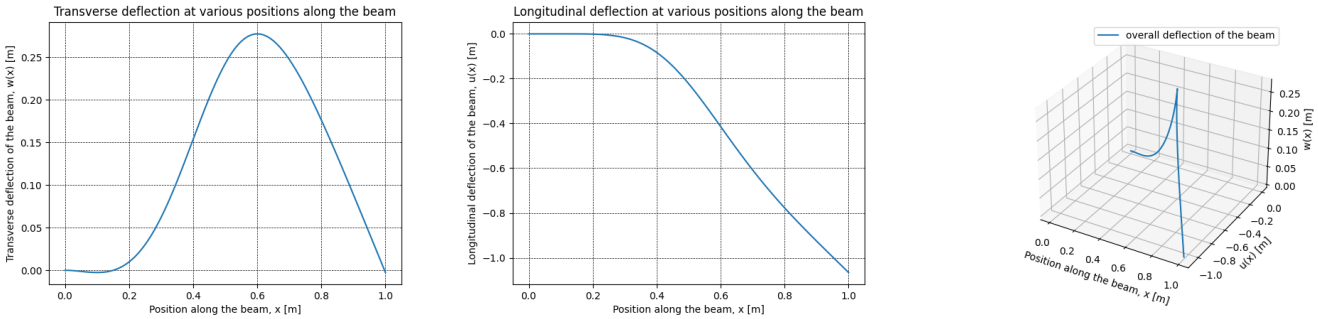

Figure 10: Transverse, longitudinal, and overall deflection of the beam using three q_i terms

2.16 Convergence towards the true solution between 2 and 3 q_i terms

Figures 9 & 10 are very similar, which implies that $\phi_h(x)$ begins to converge towards the true solution $\phi(x)$. Furthermore, we can observe that the q_1 & q_2 terms found in sections 2.12 and 2.14 are very similar, with a relative error of 1.01% & 1.42%, respectively.

2.17 Convergence towards true solution of $\psi(x)$

The convergence of $psi_h(x) \to \psi(x)$ can be observed by substituting $\psi_h(x)$ into the LHS of equation 6 below and taking its absolute value find its distance from 0 (which is the RHS of the DE). From figure 11, we see that the residual begins to decrease, indicating that $psi_h(x)$ is converging to a true solution.

$$D\psi''(x) + \int_x^L F_N \cos\left[\psi(x) - \psi(s)\right] ds = 0 \tag{6}$$

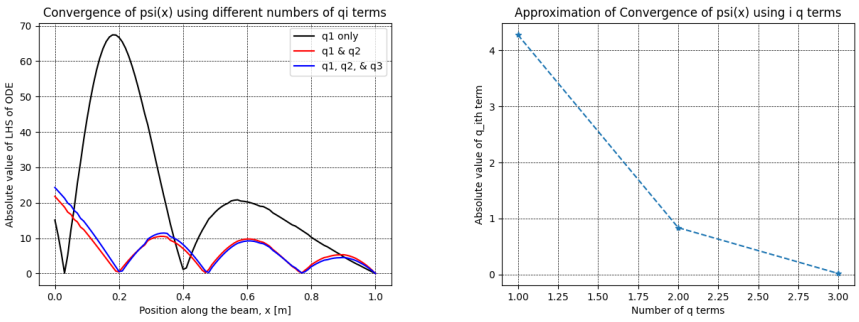

Figure 11: The convergence of $\psi_h(x) \to \psi(x)$ and the convergence of $q_i \to 0$ that may approximate the ROC of $\psi_h(x) \to \psi(x)$

The left hand side of the ODE is supposed to equal zero for all values of x $\in$ [0, L]. The residuals of the above plots of the ODEs with q_i terms where i = 1, 2, 3 can be taken to find the average and maximum error for the three cases. The residuals appear to converge; however, there is not enough information to conclude anything about the rate of convergence.

Case	**Residual (Avg.)**	**Residual (Max.)**
1D	29.6	67.4
2D	8.1	21.8
3D	8.8	24.3

Table 1: Average & maximum residual of the LHS of equation 6

There isn't much we can say about the rate of convergence with just three approximations of $\psi(x)$, we can make some inferences based on certain observations. Namely, we can see that the absolute value of the q_i^{th} term becomes successively smaller with more terms in the approximation. From this, we can infer that the rate of convergence of $\psi_h(x)$ towards $\psi(x)$ is proportional to the rate of convergence of the i^{th} q term towards 0; in other words, we are assuming that as $q_i \to 0$, $\psi_{h,i}(x) \to \psi(x)$. Plotting this assumption, we see that $\psi_h(x)$ converges towards $\psi(x)$ quadratically or sublinearly. However, it is important to note that this is just an assumption on the behaviour of $\psi_h(x)$ and we cannot conclude anything about the rate of convergence, at least not without more data.

2.18 Approximating $\psi(x)$ with $F_N(x)$ non-constant

Taking $F_N = F_N(x)$, equation 2 becomes an equation in $n+1$ variables: $\{q_1, ..., q_n, x\}$. In the 1D case, for example, we would have two variables: $q1$ and x. The most obvious way to solve this is to discretize x and to solve the system in q_i at different points; however, this is a time-consuming process that adds another order of magnitude to the speed of the solving algorithm. Another option is to evaluate the function at certain boundary conditions x, such as $x = 0$ and $x = L$. Of course, the approach also depends on the function $F_N(x)$ itself.