OPTIMIZATION-BASED SOLUTION OF THE ELASTIC BEAM EQUILIBRIUM EQUATION UNDER TANGENTIAL STRESS IN 1D USING THE FINITE ELEMENT METHOD

EMMANUEL A. AZORKO

1. Introduction

Partial differential equations (PDEs) are essential tools for modeling various physical phenomena in fields such as engineering, physics, and applied sciences. These equations describe processes like heat conduction, fluid dynamics, and structural deformation. For instance, the equilibrium of an elastic beam subjected to tangential stress is governed by the Poisson equation [2]. In one dimension, this reduces to a boundary value problem involving the second derivative of the displacement field. Specifically, for an elastic bar fixed at both ends and subjected to an external load, the governing equation is:

$$(D) \quad -u''(x) = f(x) \quad \text{for } 0 < x < 1, \quad u(0) = u(1) = 0,$$

where $u(x)$ represents the displacement of the beam, $f(x)$ is the applied load, and the boundary conditions enforce that the ends remain fixed [2].

While analytical solutions to such equations are possible for idealized cases, real-world problems often involve complex geometries, boundary conditions, or material properties, necessitating numerical methods. A traditional approach to solving PDEs numerically is the *difference method*, which approximates derivatives using finite differences over a grid [2][1]. Although effective for simpler cases, this method may encounter difficulties with irregular geometries or when higher-order continuity is required [3].

The *finite element method (FEM)* offers an alternative by reformulating the problem as an equivalent *variational formulation*, grounded in physical principles such as the minimization of potential energy. For elliptic PDEs like the Poisson equation, this reformulation results in the following minimization problem [2]:

$$\text{(M)} \quad \text{Find } u \in V \text{ such that } F(u) \leq F(v) \quad \text{for all } v \in V,$$

where V is a set of functions satisfying the boundary conditions, and $F(v)$ is a functional representing the total energy of the system. This optimization-based perspective provides both a physical interpretation and a numerical framework for solving the problem.

In FEM, the infinite-dimensional function space V is approximated by a finite-dimensional subspace V_h, spanned by piecewise polynomial basis functions [2]. This transforms the problem into a finite-dimensional minimization problem:

$$(M_h) \quad \text{Find } u_h \in V_h \text{ such that } F(u_h) \leq F(v) \quad \text{for all } v \in V_h.$$

This reformulation leads to a system of equations, which can be linear for linear PDEs or nonlinear for nonlinear PDEs.

This project investigates the use of an optimization-based FEM approach to solve the equilibrium equation of a 1D elastic beam under tangential stress. By framing the problem as

Date: December 13, 2024.

a minimization of total energy, FEM efficiently computes the displacement field while offering flexibility to address more complex configurations. The study highlights the strengths of optimization and FEM in solving practical PDEs, particularly their computational efficiency and adaptability, as well as their advantages over traditional numerical methods.

2. Variational Formulation

2.1. **Methodology.** The one-dimensional Poisson equation is

$$-u''(x) = f(x), \quad \text{for } x \in [0,1],$$

with boundary conditions $u(0) = u(1) = 0$.

In the context as discussed above, $u(x)$ represents the displacement at point x, and $f(x)$ is the external force per unit length acting on the loaded string.

By the principle of minimum potential energy of a mechanical system as our at hand, the equilibrium configuration is

$$F(u) = \frac{1}{2}\int_0^1 |u'(x)|^2\,dx - \int_0^1 f(x)u(x)\,dx.$$

The term $\frac{1}{2}\int_0^1 |u'(x)|^2 dx$ represents the strain energy stored in the system due to deformation(internal potential energy).

The term $\int_0^1 f(x)u(x)dx$ represents the work done by the external forces(external potential energy).

It can be shown that solving $-u''(x) = f(x), \quad \text{for } x \in [0,1]$ is equivalent to solving the optimization problem

$$\min_{U \in \mathbb{R}^{N-1}} F(U) = \frac{1}{2}U^T Q U - U^T b.$$

See Johnson, C.(1987) on Numerical Solution of PDEs for proof.

2.2. **Discretization Using the Finite Element Method (FEM).** We divide the interval $[0,1]$ into N elements of equal length $h = \frac{1}{N}$. We note that the nodes are located at $x_i = ih, \quad i = 0, 1, \ldots, N$.

Let us introduce the basis functions $\phi_i(x) \in V_h$, $i = 1, 2, ..., N$, defined by

$$\phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & \text{for } x \in [x_{i-1}, x_i], \\ \frac{x_{i+1} - x}{h}, & \text{for } x \in [x_i, x_{i+1}], \\ 0, & \text{otherwise}, \end{cases}$$

where $V_h \in V$ such that if $v \in V_h, v(0) = v(1) = 0$ and V is the set of all continuous functions on $[0,1]$ and first differentiable and that elements in V_h are piecewise linear when restricted on $[0,1]$

We approximate $u(x)$ as

$$u(x) \approx \sum_{j=1}^{N-1} U_j \phi_j(x), \text{where} U_j = u(x_j).$$

Substituting the approximation into the functional $F(u)$, we get

$$F(U) = \frac{1}{2}\int_0^1 \left|\left(\sum_{j=1}^{N-1} U_j \phi_j'(x)\right)\right|^2 dx - \int_0^1 f(x)\left(\sum_{j=1}^{N-1} U_j \phi_j(x)\right) dx.$$

First Term (Strain Energy):

$$\frac{1}{2}\int_0^1 \left|\left(\sum_{j=1}^{N-1} U_j \phi_j'(x)\right)\right|^2 dx = \frac{1}{2}\int_0^1 \left(\sum_{j=1}^{N-1} U_j \phi_j'(x)\right)\left(\sum_{k=1}^{N-1} U_k \phi_k'(x)\right) dx.$$

We can write this as

$$\frac{1}{2}\sum_{j=1}^{N-1}\sum_{k=1}^{N-1} U_j U_k \int_0^1 \phi_j'(x)\phi_k'(x)dx.$$

Define the entries of the stiffness matrix Q as

$$Q_{jk} = \int_0^1 \phi_j'(x)\phi_k'(x)dx.$$

Second Term (Work Done by External Forces):

$$\int_0^1 f(x)\left(\sum_{j=1}^{N-1} U_j \phi_j(x)\right) dx = \sum_{j=1}^{N-1} U_j \int_0^1 f(x)\phi_j(x)dx.$$

Define the entries of the load vector c as

$$c_j = \int_0^1 f(x)\phi_j(x)dx.$$

Combining the terms, we have

$$F(U) = \frac{1}{2}\sum_{j=1}^{N-1}\sum_{k=1}^{N-1} U_j Q_{jk} U_k - \sum_{j=1}^{N-1} U_j c_j = \frac{1}{2}U^T Q U - c^T U.$$

(Note: Since $F(U)$ is a scalar, U^TQU and U^TQ^TU are the same because Q is symmetric.)
Thus, we have expressed $F(U)$ as a quadratic function in U.
Calculating the Derivatives of Basis Functions
The derivative $\phi_j'(x)$ is constant over each element:
- For $x \in [x_{j-1}, x_j]$:

$$\phi_j'(x) = \frac{1}{h}.$$

- For $x \in [x_j, x_{j+1}]$:

$$\phi_j'(x) = -\frac{1}{h}.$$

- $\phi_j'(x) = 0$ elsewhere.

2.3. **Computing the Entries of Q.** Due to the local support of $\phi_j(x)$, Q_{jk} is non-zero only when ϕ_j and ϕ_k overlap, i.e., when $|j-k| \leq 1$.

Case 1: $j = k$

Compute Q_{jj}:

$$Q_{jj} = \int_0^1 (\phi_j'(x))^2 dx = \int_{x_{j-1}}^{x_{j+1}} (\phi_j'(x))^2 dx.$$

Since $\phi_j'(x)$ is constant on each interval:
- On $[x_{j-1}, x_j]$:

$$\phi_j'(x) = \frac{1}{h}, \quad \text{length} = h.$$

- On $[x_j, x_{j+1}]$:

$$\phi_j'(x) = -\frac{1}{h}, \quad \text{length} = h.$$

Compute

$$Q_{jj} = \left(\frac{1}{h}\right)^2 h + \left(\frac{-1}{h}\right)^2 h = \frac{1}{h} + \frac{1}{h} = \frac{2}{h}.$$

Case 2: $|j-k| = 1$

Compute $Q_{j,j+1}$ or $Q_{j+1,j}$

$$Q_{j,j+1} = \int_0^1 \phi_j'(x)\phi_{j+1}'(x) dx = \int_{x_j}^{x_{j+1}} \phi_j'(x)\phi_{j+1}'(x) dx.$$

- On $[x_j, x_{j+1}]$:
- $\phi_j'(x) = -\frac{1}{h}$. - $\phi_{j+1}'(x) = \frac{1}{h}$. - Length $= h$.

Compute

$$Q_{j,j+1} = \left(-\frac{1}{h}\right)\left(\frac{1}{h}\right) h = -\frac{1}{h}.$$

Similarly, $Q_{j+1,j} = Q_{j,j+1} = -\frac{1}{h}$.

Case 3: $|j-k| > 1$

In this case, ϕ_j and ϕ_k do not overlap, so

$$Q_{jk} = 0.$$

Therefore, Q is a tridiagonal matrix with
- Diagonal entries $Q_{jj} = \frac{2}{h}$.
- Off-diagonal entries $Q_{j,j+1} = Q_{j+1,j} = -\frac{1}{h}$.

That is

$$Q = \frac{1}{h}\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

2.4. Computing the Entries of c.

Each entry c_j is given by

$$c_j = \int_0^1 f(x)\phi_j(x)dx = \int_{x_{j-1}}^{x_{j+1}} f(x)\phi_j(x)dx.$$

Since $\phi_j(x)$ is non-zero only on $[x_{j-1}, x_{j+1}]$, we can split the integral,

$$c_j = \int_{x_{j-1}}^{x_j} f(x)\phi_j(x)dx + \int_{x_j}^{x_{j+1}} f(x)\phi_j(x)dx.$$

To compute c_j, we perform numerical integrationusinf Trapezoidal Rule on each subinterval.

First Subinterval $[x_{j-1}, x_j]$:
- $\phi_j(x) = \frac{x-x_{j-1}}{h}$. - Approximate

$$c_{j,1} = \int_{x_{j-1}}^{x_j} f(x)\phi_j(x)dx \approx \frac{h}{2}\left[f(x_{j-1})\phi_j(x_{j-1}) + f(x_j)\phi_j(x_j)\right].$$

- Note that $\phi_j(x_{j-1}) = 0$ and $\phi_j(x_j) = 1$.

So

$$c_{j,1} \approx \frac{h}{2} f(x_j).$$

Second Subinterval $[x_j, x_{j+1}]$:
- $\phi_j(x) = \frac{x_{j+1}-x}{h}$. - Approximate:

$$c_{j,2} = \int_{x_j}^{x_{j+1}} f(x)\phi_j(x)dx \approx \frac{h}{2}\left[f(x_j)\phi_j(x_j) + f(x_{j+1})\phi_j(x_{j+1})\right].$$

- $\phi_j(x_j) = 1$ and $\phi_j(x_{j+1}) = 0$.

So we have

$$c_{j,2} \approx \frac{h}{2} f(x_j).$$

Total c_j:

$$c_j = c_{j,1} + c_{j,2} = \frac{h}{2} f(x_j) + \frac{h}{2} f(x_j) = h f(x_j).$$

Thus, for each interior node x_j,

$$c_j = h f(x_j).$$

Now, we can write

$$F(U) = \frac{1}{2}U^T QU - c^T U,$$

where

- Q is the tridiagonal stiffness matrix with entries as derived.
- c is the load vector with entries $c_j = hf(x_j)$.
- U is the vector of unknown nodal values $[U_1, U_2, \ldots, U_{N-1}]^T$.

There is no constant term d in this case.

2.5. **Solving the Minimization Problem using Newton's method.** To find U that minimizes $F(U)$, we set the gradient of $F(U)$ to zero,

$$\nabla F(U) = QU - c = 0.$$

This yields the linear system:

$$QU = c.$$

Since Q is symmetric positive definite, this system has a unique solution.

For quadratic functionals, Newton's method converges in one iteration starting from any initial guess because the Hessian matrix Q is constant.

We note that Newton's Update Rule is

$$U^{(k+1)} = U^{(k)} - [\nabla^2 F(U^{(k)})]^{-1} \nabla F(U^{(k)}).$$

Since $\nabla^2 F(U) = Q$ and $\nabla F(U) = QU - b$, the update simplifies to

$$U^{(k+1)} = U^{(k)} - Q^{-1}(QU^{(k)} - b) = Q^{-1}b.$$

Thus, $U^{(k+1)} = Q^{-1}b$, independent of $U^{(k)}$. Therefore, solving $QU = b$ directly provides the solution.

2.6. **Implementation Steps.**

(1) Set Up the Mesh
- Choose N, the number of elements.
- Compute $h = \frac{1}{N}$.
- Define the nodes x_i.

(2) Assemble the Stiffness Matrix Q
- Initialize Q as an $(N-1) \times (N-1)$ zero matrix.
- Fill the diagonal entries $Q_{ii} = \frac{2}{h}$.
- Fill the off-diagonal entries $Q_{i,i+1} = Q_{i+1,i} = -\frac{1}{h}$.

(3) Assemble the Load Vector b
- For each node i, compute

$$b_i = \int_{x_{i-1}}^{x_{i+1}} f(x)\phi_i(x)dx.$$

- Use numerical integration to approximate the integrals.

(4) Solve the Linear System
- Use a linear solver (e.g., LU decomposition) to solve $QU = b$.

(5) Post-Processing

- Construct the approximate solution $u(x)$ using the nodal values U and the basis functions.
- Optionally, evaluate $u(x)$ at intermediate points for visualization.
- If $f(x)$ is chosen such that the exact solution is known compare $u(x)$ with the numerical solution.

3. Examples

To illustrate the finite element method (FEM) for solving the one-dimensional Poisson equation, consider the following example:

$$-u''(x) = \pi^2 \sin(\pi x), \quad \text{for } x \in [0, 1],$$

with boundary conditions $u(0) = u(1) = 0$.

The analytical solution to this boundary value problem is

$$u(x) = \sin(\pi x).$$

This solution satisfies both the differential equation and the boundary conditions.

4. Implementation

```
% Equation: -u''(x) = f(x), x in [0,1], with u(0)=u(1)=0
% Example: f(x) = pi^2 * sin(pi*x), exact solution u(x) = sin(pi*x)

% Parameters
N = 10;                  % Number of elements
a = 0; b = 1;            % Domain [a, b]
nodes = linspace(a, b, N+1);  % Node positions
h = (b - a)/N;           % Mesh size

%% Define the source function f(x) = pi^2 * sin(pi x)
f = @(x) (pi^2) * sin(pi * x);

% Exact Solution for Comparison
u_exact = @(x) sin(pi * x);

% Assemble the Stiffness Matrix Q
% Q is a (N-1) x (N-1) tridiagonal matrix
Q = (2/h) * diag(ones(N-1,1)) + (-1/h) * diag(ones(N-2,1),1) + (-1/h) *
    ↪ diag(ones(N-2,1),-1);

% Assemble the Load Vector c
% c is a (N-1) x 1 vector
c = zeros(N-1,1);
for i = 2:N
    x_i = nodes(i);
    c(i-1) = h * f(x_i);
end

% Solve the Linear System QU = c
U = Q \ c;
```

```
% Construct the Full Numerical Solution Including Boundary Conditions
U_full = zeros(N+1,1);
U_full(2:N) = U;

% Compute the Exact Solution at Nodes
U_exact_nodes = u_exact(nodes)';

% Compute Maximum Error
max_error = max(abs(U_full - U_exact_nodes));

% Display Maximum Error
fprintf('Maximum Error with N = %d elements: %.6f\n', N, max_error);
disp('Interior Nodal Values (U):');
for j = 1:N-1
    fprintf('u(%0.4f) = %0.6f\n', nodes(j+1), U(j));
end
disp('Full Solution Including Boundary Conditions (U_full):');
for j = 1:N+1
    fprintf('u(%0.4f) = %0.6f\n', nodes(j), U_full(j));
end

%% Visualization
x_fine = linspace(a, b, 1000);
u_exact_fine = u_exact(x_fine);

% Interpolate the numerical solution for smooth plotting
u_numerical_fine = interp1(nodes, U_full, x_fine, 'linear');

figure;
plot(x_fine, u_exact_fine, 'b-', 'LineWidth', 2); hold on;
plot(x_fine, u_numerical_fine, 'r--', 'LineWidth', 2);
plot(nodes, U_full, 'ko', 'MarkerSize', 6, 'MarkerFaceColor', 'k');
xlabel('x');
ylabel('u(x)');
title('Comparison of Exact and FEM Numerical Solutions');
legend('Exact Solution', 'FEM Numerical Solution', \\'Nodal Values', ' Location', 'Best');
grid on;
```

Listing 1. Finite Element Method Implementation for 1D Poisson Equation

5. Results

5.1. Full Solution Including Boundary Conditions. The results of the numerical solution for $N = 10$ elements are shown below, along with the exact solution $u(x) = \sin(\pi x)$ at the corresponding nodes. The maximum error is also provided.

Node (x)	**Numerical Solution** $u_h(x)$	**Exact Solution** $u(x) = \sin(\pi x)$	**Error** $\lvert u_h(x) - u(x)\rvert$
0.0000	0.000000	0.000000	0.000000
0.1000	0.311571	0.309017	0.002554
0.2000	0.592644	0.587785	0.004859
0.3000	0.815704	0.809017	0.006687
0.4000	0.958917	0.951057	0.007860
0.5000	1.008265	1.000000	0.008265
0.6000	0.958917	0.951057	0.007860
0.7000	0.815704	0.809017	0.006687
0.8000	0.592644	0.587785	0.004859
0.9000	0.311571	0.309017	0.002554
1.0000	0.000000	0.000000	0.000000

TABLE 1. Full numerical solution including boundary conditions for $N = 10$.

Maximum Error: 0.008265

6. Analysis

The finite element method (FEM) was employed to solve the one-dimensional Poisson equation governing the equilibrium of an elastic beam under tangential stress. The numerical solution obtained using $N = 10$ elements was compared against the exact analytical solution $u(x) = \sin(\pi x)$. Table 1 presents the displacement values at each node, highlighting both the numerical and exact solutions along with the corresponding errors.

6.1. **Accuracy of the Numerical Solution.** The numerical solution exhibits a high degree of accuracy, with the maximum error recorded at $x = 0.5$ being 0.008265. This small error indicates that the FEM effectively captures the behavior of the displacement field $u(x)$ across the domain. The symmetry observed in the displacement values about $x = 0.5$ aligns with the expected physical behavior of the system, further validating the FEM implementation.

6.2. **Error Distribution.** Analyzing the error distribution reveals that the errors are minimal at the boundary nodes ($x = 0$ and $x = 1$) where the displacement is fixed by the boundary conditions. The errors increase towards the center of the domain, peaking at $x = 0.5$. This trend is typical in FEM solutions, where the approximation tends to be less accurate in regions with higher curvature or where the solution exhibits rapid changes [3][1].

6.3. **Mesh Refinement and Convergence.** While $N = 10$ elements provide a reasonable approximation, increasing the number of elements N is expected to enhance the accuracy of the numerical solution. A finer mesh reduces the size of each element h, leading to a better approximation of the continuous displacement field $u(x)$. Future studies should incorporate a convergence analysis by systematically increasing N and observing the corresponding decrease in maximum error. This would confirm the theoretical convergence properties of the FEM for elliptic PDEs [2].

6.4. **Comparison with Analytical Solution.** The close agreement between the numerical and analytical solutions underscores the reliability of the FEM in solving boundary value problems involving second-order differential equations. The minimal errors observed are

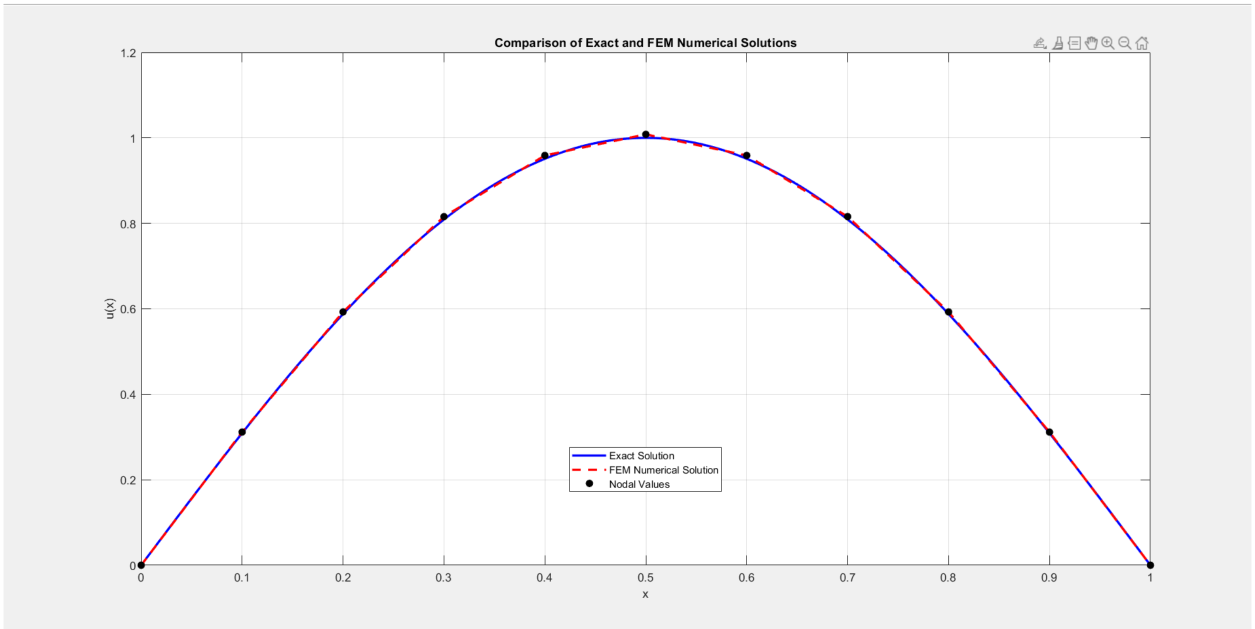

FIGURE 1. Comparison of Exact and FEM Numerical Solutions

indicative of the method's capability to accurately model physical systems, even with a moderate number of elements. This efficacy is particularly advantageous in more complex scenarios where analytical solutions are intractable.

6.5. **Computational Efficiency.** The implementation demonstrated computational efficiency, as the linear system $QU = c$ was solved directly using MATLAB's backslash operator. For larger systems with higher N, more sophisticated solvers or matrix storage techniques (e.g., sparse matrices) may be necessary to maintain computational performance. Nevertheless, the current implementation serves as a robust foundation for more extensive simulations.

6.6. **Limitations and Considerations.** While the FEM provides accurate solutions, several factors must be considered:

- **Element Size (h)**: Smaller elements yield higher accuracy but increase computational cost.
- **Basis Functions**: Higher-order basis functions can capture more complex solution behaviors but require more computational resources.
- **Boundary Conditions**: Accurate implementation of boundary conditions is crucial for the validity of the solution.
- **Source Function ($f(x)$)**: The choice of $f(x)$ affects the complexity of the integration and the overall solution accuracy.

Addressing these factors is essential for extending the FEM to more complex and higher-dimensional problems.

7. CONCLUSION

This project successfully applied the finite element method (FEM) to solve the one-dimensional Poisson equation governing the equilibrium of an elastic beam under tangential stress. By formulating the problem as an optimization task grounded in the principle of minimum potential energy, FEM provided an accurate and efficient numerical solution. The comparison between the numerical and exact analytical solutions demonstrated the method's high degree of accuracy, with minimal errors observed across all nodes.

The results underscore the FEM's robustness in handling boundary value problems, offering significant advantages over traditional numerical methods such as the finite difference method, particularly in scenarios involving complex geometries and varying material properties. The method's flexibility in accommodating different basis functions and mesh refinements further enhances its applicability to a wide range of engineering and physical problems [2].

Future work should focus on conducting a comprehensive convergence analysis by increasing the number of elements N and evaluating the corresponding impact on solution accuracy. Additionally, extending the methodology to higher-dimensional problems and exploring advanced numerical integration techniques can further solidify FEM's role as a cornerstone in numerical simulations of physical systems.

In all, the finite element method proves to be a powerful tool for solving differential equations in engineering and applied sciences, providing reliable and precise solutions essential for the design and analysis of complex structures and systems.

References

[1] I. Griva, S. Nash, & A. Sofer (2009). *Linear and Nonlinear Optimization*, 2nd ed., SIAM Press.

[2] Johnson , C.(1987). *Numerical solution of partial differential equations by finite element method.* Cambridge University Press.

[3] Slaughter, William S. (2002). *The Linearized Theory of Elasticity.*