

Advanced Finite Element Methods

Lecture Notes

Prashant K. Jha

Leslie A. Rose Department of Mechanical Engineering

South Dakota School of Mines & Technology

prashant.jha@sdsmt.edu | pjha.sci@gmail.com

Table of contents

Preface	4
Notation and Conventions	6
Typography and indices	6
Geometry and integration	6
Differential operators	7
Function spaces and variational objects	7
Finite element notation	8
Recurring physical fields	8
 I Review and Foundations	 9
1 Introduction to the Finite Element Method	10
1.1 PDE-governed engineering problems	10
1.2 From a physical model to a numerical solution	12
1.3 Why the variational formulation matters	13
1.4 Approximation and discretization	15
1.5 Assessing the reliability of a computed result	16
1.6 Computational framework used in this book	17
1.7 Scope and organization of the book	17
1.8 Key ideas	19
1.9 Exercises	19
1.10 Further reading	20
 2 Mathematical Foundations	 21
2.1 Vector spaces, norms, and inner products	21
2.2 Function spaces	21
2.3 Weak derivatives	21
2.4 Sobolev spaces	22
2.5 Linear and bilinear forms	22
 3 Partial Differential Equations and Variational Formulations	 23
3.1 Strong and weak forms	23
3.2 Abstract form	23
3.3 Energy minimization	23
 4 Model Problems in Heat Transfer and Solid Mechanics	 25
4.1 Heat conduction	25
4.2 Linear elasticity	25
4.3 Hyperelasticity	25

4.4	Common variational structure	26
5	Review of Finite Element Approximation	27
5.1	Galerkin approximation	27
5.2	Basis functions and assembly	27
5.3	Reference elements and quadrature	27
5.4	A computed figure inside the chapter	28
5.5	FEniCSx code in the same source	29
II	Error Analysis and Adaptivity	30
6	A Priori Error Estimation and Convergence	31
7	A Posteriori Error Estimation and Adaptive Finite Elements	32
III	Advanced Finite Element Formulations	33
8	Constraints in Finite Element Methods	34
9	Mixed Finite Element Methods	35
10	Nonlinear Finite Element Methods	36
11	Transient Finite Element Methods	37
IV	PDE-Constrained Optimization	38
12	PDE-Constrained Optimization	39
	References	40

Preface

These lecture notes accompany the Fall 2026 course **ME/BME 736: Advanced Finite Element Methods**. Finite element methods play a fundamental role in engineering modeling, analysis, and design. The course develops a rigorous and computationally useful understanding of the subject, using solid mechanics and heat conduction as recurring model problems. FEniCSx—an open-source, Python-based finite element library—is used throughout for implementation and numerical investigation.

The book is written in Quarto and can be rendered as both HTML and PDF, allowing mathematical developments, Python code, and computational results to be integrated in a common source. The accompanying repository collects the examples developed in the course and the book.

The book is organized into four parts:

1. **Review and Foundations**
2. **Error Analysis and Adaptivity**
3. **Advanced Finite Element Formulations**
4. **PDE-Constrained Optimization**

The first part reviews the mathematical, variational, continuum-mechanics, and finite element foundations needed for the advanced topics that follow. The second part develops a priori and a posteriori error estimation, convergence analysis, and adaptive mesh refinement. The third part addresses constraints, mixed formulations for nearly incompressible solids, nonlinear and transient finite element methods, and the associated spatial and temporal discretizations. The fourth part considers PDE-constrained optimization, including the inference of sources and model parameters, structural and topology optimization, and sensor placement. Together, these topics connect the mathematical foundations of finite element methods with tools for assessing, improving, and optimizing computational models.

The perspective of this book draws on the author’s experience applying finite element methods to deformation and fracture mechanics (Jha and Lipton 2021), multiphysics modeling (Fritz, Jha, Köppl, Oden, Wagner, et al. 2021; Fritz, Jha, Köppl, Oden, and Wohlmuth 2021), neural networks (Prashant K. Jha 2026b; Jha 2024), Bayesian inference (Jha and Oden 2022; Jha et al. 2020), and topology optimization (Galloway and Jha 2026b). The author first studied finite element methods during his master’s program at the Indian Institute of Science, Bangalore, where C. S. Jog’s lecture notes were an important resource (Jog 1978). During his doctoral studies at Carnegie Mellon University, he served for several semesters as a teaching assistant for Jacobo Bielak’s graduate introductory course on finite element methods.

The lecture notes by Douglas Arnold (Arnold 2011) and Endre Süli (Süli 2012) provide accessible treatments of the mathematical foundations of finite element methods and error analysis. More detailed discussions can be found in the books by Oden and Reddy and by Becker, Carey, and Oden (Oden and Reddy 1976; Becker et al. 1981). The late J. Tinsley Oden, with whom the author worked from 2019 to 2023 as a postdoctoral fellow and research associate, was an important influence on the perspective adopted here. Additional references will be incorporated as the chapters develop.

The book's emphasis on practical implementation also reflects the author's experience developing finite element tutorials in `CEADpx/fenics-demo` and contributing to open-source scientific software (Prashant K. Jha 2026a; Galloway and Jha 2026a; Jha 2025; Jha and Diehl 2021). The aim is to make the mathematical formulation complete in the text while using executable code to make its numerical realization transparent and reproducible.

Notation and Conventions

This section records notation used throughout the book.

Typography and indices

Object	Convention	Typical examples
Scalar	italic lowercase or Greek letter	u, T, k, ρ
Vector	bold lowercase letter	$\mathbf{x}, \mathbf{u}, \mathbf{q}$
Second-order tensor	bold uppercase or bold Greek letter	$\mathbf{F}, \mathbf{C}, \boldsymbol{\sigma}$
Fourth-order tensor	blackboard-bold uppercase letter	$\mathbb{C}$
Set or function space	calligraphic or standard space symbol	$\mathcal{T}_h, \mathcal{V}, H^1(\Omega)$
Matrix and coefficient vector	bold uppercase letter	$\mathbf{K}, \mathbf{U}$

When components are useful, Latin indices range over the spatial dimensions and repeated indices are summed unless stated otherwise. Thus

$$a \cdot b = a_i b_i, \quad (Aa)_i = A_{ij} a_j, \quad A : B = A_{ij} B_{ij}.$$

Geometry and integration

Symbol	Meaning	Symbol	Meaning
d	Spatial dimension	$\Omega \subset \mathbb{R}^d$	Open physical domain
$\partial\Omega$	Boundary of Ω	Γ_D, Γ_N	Dirichlet and Neumann portions of the boundary
$\mathbf{x}$	Point in the spatial or current configuration	$\mathbf{X}$	Point in a reference configuration
$\mathbf{n}$	Outward unit normal	dx, ds	Volume and boundary integration measures

Unless a different decomposition is stated,

$$\partial\Omega = \overline{\Gamma_D \cup \Gamma_N}, \quad \Gamma_D \cap \Gamma_N = \emptyset.$$

Differential operators

Symbol	Meaning
$\nabla u = \nabla_{\mathbf{x}} u$	Spatial gradient of a scalar field
$\nabla \mathbf{u} = \nabla_{\mathbf{x}} \mathbf{u}$	Spatial gradient of a vector field
$\nabla \cdot \mathbf{q} = \nabla_{\mathbf{x}} \cdot \mathbf{q}$	Spatial divergence of a vector field
$\nabla \cdot \boldsymbol{\sigma} = \nabla_{\mathbf{x}} \boldsymbol{\sigma}$	Spatial divergence of a second-order tensor field
$\text{Grad } \mathbf{u} = \nabla_{\mathbf{X}} \mathbf{u}$	Gradient with respect to reference coordinates
$\text{Div } \mathbf{P} = \nabla_{\mathbf{X}} \cdot \mathbf{P}$	Divergence with respect to reference coordinates

The gradient of a vector field follows the convention

$$(\nabla \mathbf{u})_{ij} = \frac{\partial u_i}{\partial x_j},$$

and the divergence of a second-order tensor follows

$$(\nabla \cdot \mathbf{A})_i = \frac{\partial A_{ij}}{\partial x_j}.$$

Uppercase operators distinguish differentiation with respect to reference coordinates from differentiation with respect to spatial coordinates.

Function spaces and variational objects

Symbol	Meaning
$L^2(\Omega)$	Square-integrable scalar functions on Ω
$H^m(\Omega)$	Functions with square-integrable weak derivatives through order m
$H_0^1(\Omega)$	H^1 functions with zero trace on the prescribed boundary
$\mathcal{U}$	Generic trial space
$\mathcal{V}$	Generic test or variational space
$\mathcal{U}_g$	Trial space satisfying prescribed Dirichlet data g
$\mathcal{V}_0$	Test space satisfying the corresponding homogeneous data
$a(u, v)$	Bilinear form
$a(u; v)$	Semilinear form (nonlinear in u and linear in v)

Symbol	Meaning
$\ell(v)$	Linear functional
$R(u; v)$	Residual evaluated in direction v
$\Pi(u)$	Energy or potential functional
$\delta\Pi(u; v)$	First variation of Π at u in direction v

The norm, seminorm, inner product, and duality pairing are written as

$$\|v\|_V, \quad |v|_{H^m(\Omega)}, \quad (u, v)_V, \quad \langle f, v \rangle.$$

Finite element notation

Symbol	Meaning
$\mathcal{T}_h$	Mesh of the domain
$K \in \mathcal{T}_h$	A mesh cell
h_K	Diameter or characteristic size of cell K
$h = \max_{K \in \mathcal{T}_h} h_K$	Global mesh-size parameter
$(K, \mathcal{P}, \mathcal{N})$	Cell, local function space, and degrees of freedom defining a finite element
$\mathcal{U}_h, \mathcal{V}_h$	Discrete trial and test spaces
u_h, v_h	Discrete trial and test functions
ϕ_i	Finite element basis function
$\mathbf{KU} = \mathbf{F}$	Representative assembled algebraic system

The book uses *cell* for a geometric member of a mesh.

Recurring physical fields

Symbol	Meaning	Symbol	Meaning
T	Temperature	$\mathbf{q}$	Heat-flux vector
$\mathbf{u}$	Displacement vector	$\boldsymbol{\varepsilon}$	Infinitesimal strain tensor
$\boldsymbol{\sigma}$	Cauchy stress tensor	$\mathbf{F}$	Deformation gradient
$\mathbf{C} = \mathbf{F}^T \mathbf{F}$	Right Cauchy–Green tensor	$\mathbf{P}$	First Piola–Kirchhoff stress tensor
$J = \det(\mathbf{F})$	Jacobian (determinant of $\mathbf{F}$)	$I_1 = \text{tr}(\mathbf{C})$	First invariant of $\mathbf{C}$ (trace)

Part I

Review and Foundations

Chapter 1

Introduction to the Finite Element Method

Engineering analysis of a continuum often begins with a mathematical model. Temperature, displacement, velocity, concentration, and electromagnetic potential are treated as fields defined over a body or region of space. Balance laws constrain how these fields can vary, constitutive equations describe the response of a particular material, and boundary and initial conditions connect the model to its surroundings. Together, these ingredients produce partial differential equations (PDEs) whose solutions represent the idealized response of the continuum.

Closed-form solutions are uncommon once geometry, material properties, or loading become realistic. A numerical method is therefore needed to replace the continuum problem by a finite calculation. The finite element method (FEM) makes this replacement through a sequence of mathematically distinct steps: it recasts the governing equations in variational form, restricts the admissible fields to a finite-dimensional space constructed over a mesh, assembles the resulting algebraic equations, and solves those equations on a computer.

The variational structure of FEM connects physical principles, approximation theory, and numerical linear algebra. It allows us to ask whether a formulation is well posed, whether a discretization is stable, how the approximation changes under mesh refinement, and what evidence supports a computed result. These questions are part of the analysis, alongside the calculation of the numerical solution.

This chapter introduces these stages without relying on the technical details developed later. It distinguishes the physical model, the PDE, the variational problem, the finite element approximation, and the algebraic system. The following chapters examine each stage in greater detail. The notation used throughout the book is collected in the [Notation and Conventions](#) reference and defined in context as it enters the analysis.

1.1 PDE-governed engineering problems

A continuum model begins by choosing the fields and mechanisms needed to answer a particular question about a physical system. Consider heat transfer in a solid body occupying a domain $\Omega \subset \mathbb{R}^d$. The primary unknown is the temperature $T(\mathbf{x}, t)$. Conservation of energy relates its rate of change to heat flux and internal heat generation. Fourier's law, the constitutive relation in this model, relates the heat flux $\mathbf{q}$ to the temperature gradient. For an isotropic material, the resulting equation has the form

$$\rho c \dot{T} - \nabla \cdot (k \nabla T) = r \quad \text{in } \Omega, \quad (1.1)$$

where ρ is density, c is specific heat, k is thermal conductivity, and r is a volumetric heat source. A

complete problem must also prescribe suitable temperature or heat-flux conditions on the boundary and an initial temperature.

Now consider a deformable solid. The primary unknown is the displacement field $\mathbf{u}(\mathbf{x})$, which describes how a material point moves from its original position. For small quasistatic deformations, the balance of linear momentum may be written schematically as

$$-\nabla \cdot \boldsymbol{\sigma} = \mathbf{b} \quad \text{in } \Omega, \quad (1.2)$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{b}$ is the body force per unit volume. Kinematics relate the displacement gradient to strain. For example, small-deformation theory uses $\boldsymbol{\varepsilon} = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$. A constitutive law then relates strain to stress; linear elasticity, for example, uses the linear relation $\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon}$. Displacements may be prescribed on one portion of the boundary and tractions on another.

Equation 1.1 and Equation 1.2 describe different physics and have unknowns of different types: temperature is a scalar field, whereas displacement is a vector-valued field. Nevertheless, the two problems share an important structure:

- a field is sought over a domain;
- a balance law governs the field in the interior;
- a constitutive law models material behavior;
- boundary and possibly initial data complete the problem;
- quantities such as heat flux, stress, or reaction force are derived from the solution.

This common structure allows one mathematical framework to treat a broad class of continuum problems. It does not make the models interchangeable. Choosing a field, balance law, constitutive response, and boundary data remains a physical modeling decision, and no discretization can correct an inappropriate model.

1.1.1 Models answer specific questions

A physical system admits many mathematical models. A thermal analysis may neglect radiation, treat conductivity as constant, or assume a steady state. A structural analysis may assume infinitesimal strain, quasistatic loading, and a linearly elastic material. Each assumption limits the conditions under which the model is appropriate. For example, a model based on linearized strain and small deformations is generally unsuitable for rubbers and other soft materials undergoing large deformations.

Suppose that a metal support is heated along one surface and fixed mechanically along another. A thermal model may predict temperature and heat flux while saying nothing about thermal stress. A linear elastic model may predict small displacements under moderate loading but become inappropriate after large rotations or plastic deformation. These are limitations of the model, and mesh refinement cannot remove them. If observations include stresses at several points, a mechanical model that omits thermal effects may fail to represent the data even when its finite element discretization is highly accurate.

For this reason, a finite element calculation should begin with an explicit statement of:

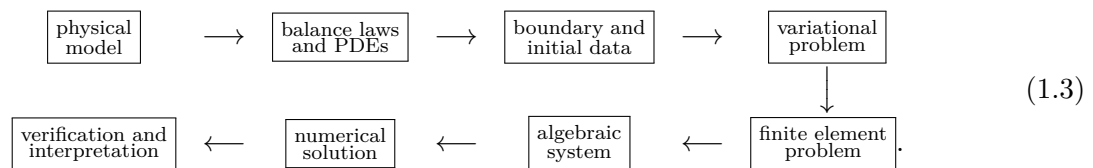
1. the domain and relevant geometric idealizations;
2. the primary unknown fields;
3. the balance laws and constitutive assumptions;

4. boundary and initial conditions;
5. parameters and source terms;
6. the output quantities, or quantities of interest (QoIs), needed from the solution.

This statement separates model construction from numerical solution and gives a basis for interpreting the result later.

1.2 From a physical model to a numerical solution

The book organizes each model problem through the following sequence:



Each box in Equation 1.3 denotes a distinct stage of modeling or computation.

1.2.1 Physical model

The physical model identifies the idealized system, its state variables, and the mechanisms retained in the analysis. Material response, source terms, and loading assumptions enter at this level. At this stage, the analyst specifies what is being modeled and the assumptions under which the model will be used. The governing differential equations follow from those choices.

1.2.2 Balance laws, constitutive equations, and PDEs

Balance laws express conservation of quantities such as energy, mass, or linear momentum. Constitutive equations close the system by relating fluxes or stresses to the primary fields. Combining these statements produces governing PDEs such as Equation 1.1 and Equation 1.2. The resulting equations are usually called the *strong form* when they are required to hold pointwise and all derivatives are interpreted classically.

1.2.3 Boundary and initial data

A differential equation alone is not enough to obtain a unique solution. Boundary conditions describe interaction with the environment: a temperature may be prescribed, a heat flux imposed, a displacement constrained, or a traction applied. Time-dependent problems also require initial data. These conditions are part of the mathematical model, not implementation details to be added after discretization.

1.2.4 Variational problem

The variational problem asks that the governing equations hold through their action on an admissible class of test functions. Derivatives are redistributed using integration by parts, and boundary terms

become part of the formulation. The unknown is still a function drawn from an infinite-dimensional space. No finite-dimensional approximation has been introduced at this stage. Under appropriate assumptions, the variational problem is equivalent to the strong form, but it admits solutions with weaker differentiability. Such a solution is called a *weak solution* to distinguish it from a classical, or strong, solution of the strong form.

1.2.5 Finite element problem

The finite element problem restricts the trial and test functions to finite-dimensional spaces constructed over a mesh. This is the approximation step. Mesh geometry, polynomial order, interelement continuity, and the treatment of essential boundary data determine the discrete spaces.

1.2.6 Algebraic system

After finite element discretization, the variational problem becomes a system of algebraic equations for the coefficients associated with the finite element degrees of freedom. A linear scalar PDE may produce

$$\mathbf{K}\mathbf{U} = \mathbf{F}, \quad (1.4)$$

where $\mathbf{U}$ contains coefficients of the finite element field. Nonlinear, mixed, constrained, or transient formulations lead to different algebraic structures. The coefficient vector is a representation of the discrete field; it is not the continuum solution itself.

1.2.7 Numerical solution

An algebraic solver computes an approximation to the coefficient vector. Direct and iterative solvers, preconditioners, stopping tolerances, and floating-point arithmetic enter at this stage. Solving the algebraic equations inaccurately can obscure the behavior of an otherwise stable discretization.

1.2.8 Verification and interpretation

The computed field must be checked and interpreted. Verification asks whether the equations were discretized and solved correctly. Interpretation asks whether derived quantities are meaningful for the original question. Keeping these levels separate prevents common reasoning errors. A solver failure is not automatically a defect in the PDE, a mesh-refinement trend does not validate a constitutive model, and agreement with the assembled equations does not prove agreement with the continuum problem.

1.3 Why the variational formulation matters

The finite element method is naturally built from a variational formulation rather than directly from pointwise differential equations. To see why, consider the representative scalar problem

$$-\nabla \cdot (k \nabla u) = f \quad \text{in } \Omega. \quad (1.5)$$

The differential operator contains second derivatives of u when k is smooth. A classical solution must possess enough pointwise differentiability for those derivatives to exist and satisfy the equation. Such regularity can fail near corners, across material interfaces, or where boundary conditions change.

In a variational formulation, the equation is multiplied by a test function and integrated over the domain. Integration by parts transfers one derivative from the unknown to the test function and exposes a boundary term. The resulting statement has the abstract form

$$\text{find } u \in \mathcal{U}_g \text{ such that } a(u, v) = \ell(v) \quad \text{for every } v \in \mathcal{V}_0. \quad (1.6)$$

Here $\mathcal{U}_g$ is a trial space containing fields that satisfy prescribed values on the Dirichlet boundary, and $\mathcal{V}_0$ is the associated test space with homogeneous boundary values. The form $a(u, v)$ contains the differential operator after integration by parts, while $\ell(v)$ contains sources and prescribed fluxes or tractions. Chapter 3 constructs this statement carefully.

Several important consequences are already visible.

1.3.1 Weaker differentiability requirements

The variational problem can admit fields whose derivatives exist only in the weak sense. Weak derivatives and Sobolev spaces provide the precise language for this enlargement of the admissible class. Chapter 2 develops that language. This broader solution space is important because piecewise polynomial finite element functions are generally not globally smooth enough to satisfy a second-order PDE pointwise, but they can satisfy the corresponding variational equation.

1.3.2 Boundary conditions enter in different ways

Prescribed values are built into the admissible trial space and are called *essential* boundary conditions. Fluxes and tractions naturally arise in boundary integrals and are called *natural* boundary conditions. This distinction follows from the variational structure.

1.3.3 Physical principles appear naturally

In mechanics, the test function can be interpreted as a virtual displacement, and the variational equation expresses virtual work. In symmetric conservative problems, the same solution may make an energy functional stationary or minimize it over an admissible set. A variational formulation and an energy principle are closely related in those cases, but they are not synonymous: many useful variational problems do not arise from minimization.

1.3.4 Properties of the variational problem

The form $a(u, v) = \ell(v)$ can be examined for properties such as symmetry, continuity, and coercivity. For a conforming Galerkin method, appropriate continuous properties are inherited by the finite element problem. These properties govern existence, uniqueness, and stability and later help determine the quality of the finite element approximation.

1.4 Approximation and discretization

The variational problem remains infinite dimensional: its unknown is a function, and the equation must hold for every admissible test function. Computation requires finite-dimensional trial and test spaces. FEM creates those spaces from a mesh and local approximation rules.

1.4.1 Meshes and local approximation

A mesh $\mathcal{T}_h$ partitions the domain into nonoverlapping elements K . Typical elements include intervals in one dimension, triangles and quadrilaterals in two dimensions, and tetrahedra and hexahedra in three dimensions. On each element, the unknown is represented by a simple local function, usually a polynomial. Neighboring representations are connected so that the assembled field has the interelement continuity required by the variational problem. Discontinuous Galerkin methods relax this continuity requirement and instead couple neighboring elements through interface terms (Arnold et al. 2002). This book focuses on conforming finite element methods; discontinuous Galerkin methods are outside its present scope.

Each geometric member of the mesh is called a *cell* or *element*. A finite element is more precisely the combination of an element, a local function space, and a set of degrees of freedom.

1.4.2 Finite-dimensional spaces

Let $\mathcal{V}_h$ denote a finite-dimensional test space constructed over $\mathcal{T}_h$. A conforming method chooses $\mathcal{V}_h \subset \mathcal{V}_0$, so every discrete test function is also admissible for the continuous variational problem. The corresponding Galerkin statement has the schematic form

$$\text{find } u_h \in \mathcal{U}_h \text{ such that } a(u_h, v_h) = \ell(v_h) \quad \text{for every } v_h \in \mathcal{V}_h. \quad (1.7)$$

The subscript h indicates dependence on the mesh. It does not mean that the discrete field is simply the exact field evaluated at grid points. The finite element solution is determined by the discrete variational equation over an entire function space.

1.4.3 Local-to-global construction

Finite element basis functions have local support: each is nonzero only on a small neighborhood of elements. Integrals can therefore be evaluated element by element and accumulated into global matrices and vectors. Two basis functions that do not overlap make no joint contribution, which produces sparse algebraic systems.

This local-to-global construction is one of the defining computational features of FEM. It permits complex domains to be assembled from standard element types, allows material data to vary by region, and makes refinement possible in selected parts of the mesh. It also permits distributed computing, in which different processors assemble contributions from the portions of the mesh they own.

1.4.4 Reference elements and geometric flexibility

Calculations on many physical elements are organized by mapping them from a small collection of reference elements. Shape functions, quadrature rules, and derivatives can be defined once on a reference element and transformed to each physical element. When the geometry and the unknown field are represented using the same type of basis, the construction is called *isoparametric*.

These ideas explain how FEM accommodates curved boundaries and complicated geometries while retaining a systematic implementation. Chapter 5 develops the finite-element triple, Lagrange spaces, element mappings, quadrature, assembly, and discrete boundary conditions in detail.

1.5 Assessing the reliability of a computed result

A completed solve establishes that an algebraic procedure returned a result. Further checks are needed to determine whether that result approximates the intended physical system with acceptable accuracy. The following sources of discrepancy should be considered separately.

Source	Question
Modeling error	Do the physical assumptions represent the system of interest?
Data uncertainty	Are geometry, material parameters, loads, and boundary data known adequately?
Discretization error	Does the finite-dimensional space approximate the continuum solution sufficiently well?
Quadrature error	Were element and boundary integrals evaluated accurately enough?
Algebraic error	Were the discrete equations solved to an adequate tolerance?
Implementation error	Does the code represent the intended formulation?
Interpretation error	Are reported outputs consistent with the model and discretization?

The distinction between *verification* and *validation* is useful. Verification asks whether the mathematical model has been solved correctly. Validation asks whether that model is an adequate representation of the physical system for the intended use. Mesh refinement, exact solutions, manufactured solutions, conservation checks, and independent implementations support verification. Comparison with experimental or trusted observational data supports validation.

1.5.1 Interpreting refinement studies

Under the assumptions required for convergence, refining the mesh or increasing the polynomial order should bring a consistent and stable discretization closer to the continuum solution. A refinement study provides useful numerical evidence, but its results require careful interpretation. A poor model can converge, an incorrectly implemented method can show an accidental plateau, and some quantities converge more slowly than the primary field.

Later chapters make refinement studies quantitative. A priori analysis relates the approximation error to mesh size, polynomial order, solution regularity, and stability. A posteriori analysis uses the computed solution and problem data to estimate where the discretization error is concentrated. Adaptive algorithms then use this information to distribute computational effort.

1.5.2 Derived quantities require separate attention

The primary unknown may be temperature or displacement, while the engineering quantity of interest is heat flux, stress, reaction force, average temperature, or compliance. Such outputs involve derivatives, boundary reactions, or integrals and need not inherit the same accuracy as the primary field. A credible analysis identifies the quantity of interest before choosing the mesh and verification strategy.

1.6 Computational framework used in this book

The computational material follows the same order as the mathematics: model, variational form, discrete spaces, algebraic problem, solver, and verification.

1.6.1 Python and Jupyter

Python provides the common language for numerical examples, preprocessing, postprocessing, and verification. Jupyter allows executable computations, figures, and explanatory text to remain together. Short examples appear directly in the chapters; longer investigations belong in notebooks, and reusable implementations belong in source modules or examples.

1.6.2 FEniCSx and UFL

FEniCSx represents meshes, finite element spaces, functions, boundary conditions, and variational forms as computational objects. The Unified Form Language (UFL) expresses integrals using notation close to their mathematical form. From those forms, the software generates element-level computations and assembles global algebraic objects.

1.6.3 PETSc and scalable algebra

The assembled matrices and vectors are managed through PETSc. PETSc supplies linear and nonlinear solvers, preconditioners, distributed data structures, and parallel algorithms.

1.7 Scope and organization of the book

The book begins with a focused review and then develops the analytical and computational tools required for advanced formulations.

1.7.1 Review and foundations

Chapter 2 introduces vector and tensor operations, function spaces, weak derivatives, Sobolev spaces, norms, bilinear forms, and well-posedness. Chapter 3 derives variational formulations from boundary-value problems and explains trial and test spaces, essential and natural boundary conditions, and energy stationarity. Chapter 4 develops heat conduction, linear elasticity, and the finite-deformation quantities needed for hyperelasticity. Chapter 5 constructs finite element spaces and connects meshes, basis functions, mappings, quadrature, assembly, boundary conditions, and FEniCSx.

1.7.2 Error analysis and adaptivity

A priori analysis relates stability and approximation properties to convergence. A posteriori analysis seeks computable information about the error in a completed solution. Adaptive FEM uses that information to refine or enrich the discretization where additional resolution is most valuable.

1.7.3 Constraints and mixed formulations

Constraints arise in contact, incompressibility, and other admissibility conditions. Mixed methods introduce multiple primary fields, such as displacement and pressure or flux and temperature. Their stability depends on compatibility between spaces and cannot be inferred from the behavior of a single scalar coercive problem.

1.7.4 Nonlinear and transient problems

Finite-deformation mechanics produces nonlinear residual equations. Their solution requires consistent linearization, tangent operators, and iterative methods. Transient problems add time discretization and require separate control of spatial and temporal errors.

1.7.5 PDE-constrained optimization

Optimization treats a PDE solution as a constraint while adjusting design or control variables to improve an objective. Adjoint equations make sensitivities computationally tractable and bring together variational formulations, linearized operators, solver design, and quantities of interest.

Across all these topics, the same questions recur:

1. What is the physical and mathematical problem?
2. Which trial and test spaces express its structure?
3. Is the continuous problem well posed?
4. Is the discrete formulation stable and consistent?
5. What controls the approximation error?
6. How should the algebraic equations be solved?
7. What evidence supports the reported result?

The chapters return to these questions when introducing each formulation and its computational implementation.

1.8 Key ideas

- A continuum model, its strong PDE, its variational formulation, its finite element approximation, and its algebraic system are distinct mathematical objects.
- Boundary and initial conditions are part of the model, not postprocessing or software details.
- Variational formulations reduce differentiability requirements, expose natural boundary terms, and provide the structure used by FEM.
- Finite element spaces combine local approximation on mesh elements with a global continuity structure.
- Local support leads to elementwise computation and sparse global systems.
- A computed solution is credible only when modeling, discretization, algebra, implementation, and interpretation errors are considered separately.
- Computation should follow the mathematical formulation and include explicit verification evidence.

1.9 Exercises

1. A rectangular plate $\Omega = (0, L) \times (0, H)$ has constant thermal conductivity k and a volumetric heat source r . The temperature is prescribed as T_0 on the left edge, the normal heat flux is prescribed as $\bar{q}$ on the right edge, and the top and bottom edges are insulated. Without solving the problem, state:
 - a. the primary unknown and its type (scalar, vector, or tensor);
 - b. Fourier's constitutive law for $\mathbf{q}$;
 - c. the steady governing PDE;
 - d. the boundary condition on each edge, classified as essential or natural;
 - e. three distinct QoIs that could be computed from the solution.
2. A two-dimensional cantilever of length L and height H is clamped at $x = 0$ and carries a prescribed downward traction at $x = L$. It is modeled using small-strain linear elasticity. Prepare a table containing two modeling assumptions, two discretization choices, and two verification checks. For each entry, state whether mesh refinement can address it and justify the answer in one sentence.
3. Classify each item below as belonging primarily to the physical model, variational problem, finite element approximation, algebraic system, or numerical solver. Give a one-sentence justification for each classification:
 - a. a prescribed traction;
 - b. a test function;
 - c. a quadratic triangular element;
 - d. a Krylov-method stopping tolerance;
 - e. a coefficient vector $\mathbf{U}$;
 - f. a constitutive tensor $\mathbb{C}$.
4. Let u_h be continuous and piecewise linear on a conforming triangular mesh. State where its classical second derivatives exist and explain why u_h generally cannot satisfy Equation 1.5 pointwise across element boundaries. Then explain, using the integration-by-parts argument from this chapter, why u_h can still be admissible in the corresponding variational problem.
5. A finite element analysis of the cantilever in Exercise 2 produces a smooth displacement plot, but the reported QoI is the average normal stress over a specified region near the clamp. Design

- a mesh-refinement study for this QoI. State the sequence of meshes to be compared, the value to record on each mesh, and a numerical criterion for deciding that the QoI is sufficiently resolved. Explain why visual smoothness of the displacement field is not that criterion.
6. A tensile coupon is tested in a laboratory, and the same test is represented by a structural finite element model. Give two verification activities and two validation activities for this study. For each activity, identify the reference result or data needed and the discrepancy that would be measured.
 7. For each item below, identify the box in Equation 1.3 where it first enters and state one error or uncertainty it may introduce:
 - a. calibration of a material parameter from test data;
 - b. selection of a numerical quadrature rule;
 - c. local mesh refinement;
 - d. selection of a linear solver and stopping tolerance;
 - e. comparison of a computed QoI with experimental measurements.

1.10 Further reading

Broad treatments of the mathematical and variational foundations of FEM are available in (Oden and Reddy 1976; Süli 2012). Mechanics-oriented developments and computational constructions can be found in (Jog 1978; Becker et al. 1981), while (Larson and Bengzon 2010; Arnold 2011) provide further perspectives on approximation, implementation, stability, and error assessment. Oden and Demkowicz provide the functional-analysis background used in this book (Oden and Demkowicz 2018). Discontinuous Galerkin methods are outside the present scope; Arnold et al. provide a unified analysis of these methods for elliptic problems (Arnold et al. 2002).

Chapter 2

Mathematical Foundations

This chapter establishes only the functional-analytic language required later. The treatment is intentionally selective. Süli's notes provide a concise path from function spaces to weak derivatives and Sobolev spaces, while Oden and Demkowicz provide the broader functional-analysis framework (Süli 2012; Oden and Demkowicz 2018).

2.1 Vector spaces, norms, and inner products

Definition 2.1 (Norm). Let V be a vector space over $\mathbb{R}$. A mapping $\|\cdot\|_V : V \rightarrow \mathbb{R}$ is a **norm** if it is positive definite, absolutely homogeneous, and satisfies the triangle inequality.

The associated notion of convergence is

$$v_n \rightarrow v \text{ in } V \iff \|v_n - v\|_V \rightarrow 0.$$

2.2 Function spaces

For finite element analysis the most important examples are

$$L^2(\Omega), \quad H^1(\Omega), \quad H_0^1(\Omega).$$

The L^2 inner product is

$$(u, v)_{L^2(\Omega)} = \int_{\Omega} uv \, dx,$$

with induced norm

$$\|u\|_{L^2(\Omega)} = \left(\int_{\Omega} |u|^2 \, dx \right)^{1/2}.$$

2.3 Weak derivatives

The weak derivative is motivated directly by integration by parts. For a sufficiently smooth u and a compactly supported smooth test function v ,

$$\int_{\Omega} \frac{\partial u}{\partial x_i} v \, dx = - \int_{\Omega} u \frac{\partial v}{\partial x_i} \, dx.$$

This identity can be used as the definition of the derivative even when the classical derivative does not exist pointwise, which is the key step toward Sobolev spaces (Süli 2012).

2.4 Sobolev spaces

For the second-order problems used repeatedly in this course,

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) : \frac{\partial v}{\partial x_i} \in L^2(\Omega), \, i = 1, \dots, d \right\}.$$

The associated norm is

$$\|v\|_{H^1(\Omega)}^2 = \|v\|_{L^2(\Omega)}^2 + \|\nabla v\|_{L^2(\Omega)}^2. \quad (2.1)$$

Equation 2.1 will be used repeatedly when discussing well-posedness and finite element error.

2.5 Linear and bilinear forms

A large class of variational problems can be written as

$$a(u, v) = \ell(v) \quad \forall v \in V,$$

where $a : V \times V \rightarrow \mathbb{R}$ is a bilinear form and $\ell : V \rightarrow \mathbb{R}$ is a linear functional.

Theorem 2.1 (Lax–Milgram theorem). *If V is a Hilbert space, $a(\cdot, \cdot)$ is continuous and coercive on $V \times V$, and ℓ is a continuous linear functional on V , then the variational problem has a unique solution $u \in V$.*

The theorem is stated here for later use. Its proof is not needed for the review portion of the course.

! What students should retain

The purpose of this chapter is not abstract functional analysis for its own sake. The spaces specify what regularity we require from trial and test functions, while continuity and coercivity are the properties that later control well-posedness and finite element error.

Chapter 3

Partial Differential Equations and Variational Formulations

The finite element method is built from a variational problem rather than directly from the strong differential equation. Jog develops this transition from an engineering mechanics viewpoint, while Süli and Arnold formulate it in the language of weak solutions and Hilbert spaces (Jog 1978; Süli 2012; Arnold 2011).

3.1 Strong and weak forms

Consider the scalar diffusion problem

$$-\nabla \cdot (k \nabla u) = f \quad \text{in } \Omega,$$

with suitable boundary conditions. Multiplication by a test function v and integration by parts give the weak form

$$\int_{\Omega} k \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx + \text{boundary terms}.$$

The boundary terms distinguish essential and natural boundary conditions.

3.2 Abstract form

After defining the admissible trial and test spaces, the problem is written as

$$\text{find } u \in V \quad \text{such that} \quad a(u, v) = \ell(v) \quad \forall v \in V.$$

3.3 Energy minimization

For a symmetric coercive bilinear form, the same solution minimizes

$$\Pi(v) = \frac{1}{2} a(v, v) - \ell(v).$$

The distinction between a general variational equation and an energy minimization problem will matter later for nonlinear and constrained problems.

Chapter 4

Model Problems in Heat Transfer and Solid Mechanics

Heat conduction and solid mechanics are used throughout the course because they expose the same finite element machinery through different physics. Larson and Bengzon treat both within a common finite element framework, while Arnold derives the heat equation directly from balance and Fourier conduction (Larson and Bengzon 2010; Arnold 2011).

4.1 Heat conduction

For temperature $T(\mathbf{x}, t)$, conservation of energy and Fourier's law lead to

$$\rho c \dot{T} - \nabla \cdot (k \nabla T) = q.$$

The steady problem is obtained by setting $\dot{T} = 0$.

4.2 Linear elasticity

For displacement $\mathbf{u}$, the infinitesimal strain tensor is

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T).$$

The quasistatic balance equation is

$$-\nabla \cdot \boldsymbol{\sigma} = \mathbf{f},$$

with constitutive relation $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}(\mathbf{u})$ for linear elasticity.

4.3 Hyperelasticity

For finite deformation, the deformation gradient is

$$\mathbf{F} = \mathbf{I} + \text{Grad } \mathbf{u},$$

and a hyperelastic material is described through a stored-energy density $W(\mathbf{F})$. The nonlinear variational problem will be revisited in the nonlinear finite element module.

4.4 Common variational structure

Despite their different physical meanings, the problems above can be organized through a residual statement

$$R(u; v) = 0 \quad \forall v \in V.$$

This common structure is the reason the same finite element software framework can treat heat transfer, linear elasticity, and nonlinear solid mechanics.

Chapter 5

Review of Finite Element Approximation

Jog's abstract formulation emphasizes the sequence from a finite-dimensional subspace to assembly, boundary conditions, solution, and postprocessing. Larson and Bengzon provide a complementary approximation and implementation viewpoint (Jog 1978; Larson and Bengzon 2010).

5.1 Galerkin approximation

Given a variational problem on V , choose a finite-dimensional subspace

$$\mathcal{V}_h \subset V.$$

The conforming Galerkin approximation is

$$\text{find } u_h \in \mathcal{V}_h \quad \text{such that} \quad a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in \mathcal{V}_h. \quad (5.1)$$

5.2 Basis functions and assembly

Let $\{\phi_i\}_{i=1}^N$ be a basis for $\mathcal{V}_h$ and write

$$u_h = \sum_{j=1}^N U_j \phi_j.$$

Substitution into Equation 5.1 yields

$$\mathbf{KU} = \mathbf{F},$$

where

$$K_{ij} = a(\phi_j, \phi_i), \quad F_i = \ell(\phi_i).$$

5.3 Reference elements and quadrature

The same implementation is reused over many physical elements by mapping from a reference element. Numerical quadrature is then used to evaluate element integrals.

5.4 A computed figure inside the chapter

The following cell is intentionally simple. Its purpose is to show how a computed result appears directly in the source and receives a cross-reference.

```
import numpy as np
import matplotlib.pyplot as plt

h = np.array([0.5, 0.25, 0.125, 0.0625])
error = 0.8 * h

plt.figure()
plt.loglog(h, error, "o-")
plt.xlabel(r"mesh size $h$")
plt.ylabel("error")
plt.grid(True, which="both")
plt.tight_layout()
plt.show()
```

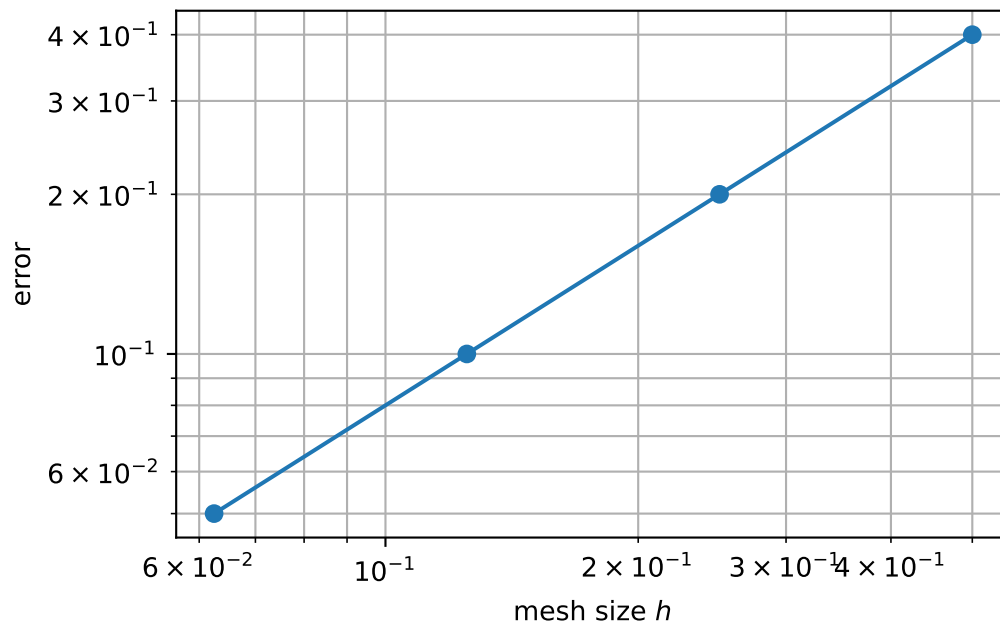


Figure 5.1: Illustrative first-order convergence curve generated during the book build.

The figure can then be referenced normally as Figure 5.1. In the actual error-analysis chapter, the values will come from FEniCSx computations rather than from an illustrative array.

5.5 FEniCSx code in the same source

A FEniCSx cell can be executable once the course environment is installed. It is marked `eval: false` here so that this template remains portable.

```
1 from mpi4py import MPI
2 from dolfinx import fem, mesh
3
4 # Mesh and scalar Lagrange space
5 domain = mesh.create_unit_square(MPI.COMM_WORLD, 32, 32)
6 V = fem.functionspace(domain, ("Lagrange", 1))
```

The important design choice is that the mathematical object V_h is introduced first, and the FEniCSx object is then presented as its computational realization.

Example 5.1 (Mathematical objects and FEniCSx objects). A typical chapter will explicitly connect the mesh $\mathcal{T}_h$, finite element space V_h , trial and test functions, variational forms, boundary conditions, and solver objects to their FEniCSx counterparts.

Part II

Error Analysis and Adaptivity

Chapter 6

A Priori Error Estimation and Convergence

Chapter placeholder. Content will be developed from the finalized course notes and references.

Chapter 7

A Posteriori Error Estimation and Adaptive Finite Elements

Chapter placeholder. Content will be developed from the finalized course notes and references.

Part III

Advanced Finite Element Formulations

Chapter 8

Constraints in Finite Element Methods

Chapter placeholder. Content will be developed from the finalized course notes and references.

Chapter 9

Mixed Finite Element Methods

Chapter placeholder. Content will be developed from the finalized course notes and references.

Chapter 10

Nonlinear Finite Element Methods

Chapter placeholder. Content will be developed from the finalized course notes and references.

Chapter 11

Transient Finite Element Methods

Chapter placeholder. Content will be developed from the finalized course notes and references.

Part IV

PDE-Constrained Optimization

Chapter 12

PDE-Constrained Optimization

Chapter placeholder. Content will be developed from the finalized course notes and references.

References

- Arnold, Douglas N. 2011. “Lecture Notes on Numerical Analysis of Partial Differential Equations.” *Numerical Analysis of Differential Equations, Version of*, 09–05. <https://www-users.cse.umn.edu/~arnold/8445.f12/notes.pdf>.
- Arnold, Douglas N., Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. 2002. “Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems.” *SIAM Journal on Numerical Analysis* 39 (5): 1749–79. <https://doi.org/10.1137/S0036142901384162>.
- Becker, Eric B., Graham F. Carey, and J. Tinsley Oden. 1981. *Finite Elements: An Introduction*. Prentice-Hall.
- Fritz, Marvin, Prashant K Jha, Tobias Köppl, J Tinsley Oden, Andreas Wagner, and Barbara Wohlmuth. 2021. “Modeling and Simulation of Vascular Tumors Embedded in Evolving Capillary Networks.” *Computer Methods in Applied Mechanics and Engineering* 384: 113975.
- Fritz, Marvin, Prashant K Jha, Tobias Köppl, J Tinsley Oden, and Barbara Wohlmuth. 2021. “Analysis of a New Multispecies Tumor Growth Model Coupling 3D Phase-Fields with a 1D Vascular Network.” *Nonlinear Analysis: Real World Applications* 61: 103331.
- Galloway, Ian, and Prashant K Jha. 2026a. *CEADpx/Top_optim - Joint Material-Topology Optimization Framework for Field-Responsive Soft Materials*. Zenodo, released July. <https://doi.org/10.5281/zenodo.21266889>.
- Galloway, Ian, and Prashant K Jha. 2026b. “Model-Informed Joint Material-Structural Optimization of Hard-Magnetic Soft Materials.” *arXiv Preprint arXiv:2607.14397*, ahead of print. <https://doi.org/10.48550/arXiv.2607.14397>.
- Jha, PK, and R Lipton. 2021. “Finite Element Approximation of Nonlocal Dynamic Fracture Models.” *Discrete and Continuous Dynamical Systems-Series B* 26 (3): 1675.
- Jha, Prashant K. 2024. “Residual-Based Error Corrector Operator to Enhance Accuracy and Reliability of Neural Operator Surrogates of Nonlinear Variational Boundary-Value Problems.” *Computer Methods in Applied Mechanics and Engineering* 419: 116595.
- Jha, Prashant K. 2026a. *CEADpx: Neural_operators*. Version survey26_v2. Zenodo. <https://doi.org/10.5281/zenodo.21197314>.
- Jha, Prashant K. 2025. “PeriDEM – High-Fidelity Modeling of Granular Media Consisting of Deformable Complex-Shaped Particles.” *Journal of Open Source Software* 10 (116): 7525. <https://doi.org/10.21105/joss.07525>.

- Jha, Prashant K. 2026b. “From Theory to Application: A Practical Introduction to Neural Operators in Scientific Computing.” *Mathematics* 14 (13): 2421. <https://doi.org/10.3390/math14132421>.
- Jha, Prashant K, Lianghao Cao, and J Tinsley Oden. 2020. “Bayesian-Based Predictions of COVID-19 Evolution in Texas Using Multispecies Mixture-Theoretic Continuum Models.” *Computational Mechanics* 66 (5): 1055–68.
- Jha, Prashant K., and Patrick Diehl. 2021. “NLMech: Implementation of Finite Difference/Meshfree Discretization of Nonlocal Fracture Models.” *Journal of Open Source Software* 6 (65): 3020. <https://doi.org/10.21105/joss.03020>.
- Jha, Prashant K., and J. Tinsley Oden. 2022. “Goal-Oriented a-Posteriori Estimation of Model Error as an Aid to Parameter Estimation.” *Journal of Computational Physics* 470: 111575. <https://doi.org/10.1016/j.jcp.2022.111575>.
- Jog, C. S. 1978. “Introduction to the Finite Element Method.” *ME257 Course Notes, Department of Mechanical Engineering, Indian Institute of Science (IISc), Bangalore*. <https://mecheng.iisc.ac.in/cs jog/me257Notes/fem.pdf>.
- Larson, Mats G., and Fredrik Bengzon. 2010. *The Finite Element Method: Theory, Implementation, and Practice*. Springer.
- Oden, J. Tinsley, and Leszek F. Demkowicz. 2018. *Applied Functional Analysis*. 3rd ed. Chapman; Hall/CRC.
- Oden, J. Tinsley, and J. N. Reddy. 1976. *An Introduction to the Mathematical Theory of Finite Elements*. Wiley.
- Süli, Endre. 2012. *Lecture Notes on Finite Element Methods for Partial Differential Equations*. Mathematical Institute, University of Oxford. <https://people.maths.ox.ac.uk/suli/fem.pdf>.