2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #51385

Industry Perspective on Finite Element Analysis Curriculum

Dr. Alex C. Szatmary, Cadence Design Systems

Alex Szatmary is an academic consulting engineer at MSC Software, part of Cadence Design Systems, supporting students and faculty in use of engineering software. He creates instructor resources, videos, and software tools to make it easier to teach engineering simulation, with a focus on finite element analysis and dynamics. He previously started a mechanical engineering program, taught almost all ME courses, and led ME efforts to successfully obtain ABET accreditation.

©American Society for Engineering Education, 2026

Industry Perspective on Finite Element Analysis Curriculum

Abstract

Finite element analysis (FEA) is the leading method for structural simulation. Introductory FEA courses for advanced undergraduates usually focus on the numerical skills needed for developers of FEA solvers, rather than the practical skills needed by structural analysts to use FEA responsibly. Because students are much more likely to become analysts than developers, typical FEA curriculum is not aligned with industry practice. Perhaps focusing on the mathematical fundamentals is intended to give a depth of knowledge needed for professional practice—but there is not clear evidence that this depth is needed or is useful in practice. To show another perspective on how people can learn FEA, this manuscript is a review of various sources used for training engineers in industry. We consider sources including certifications, books and courses for analysts in industry, and solver manuals as evidence of what analysts should be able to do. The NAFEMS Professional Simulation Engineer Computational Structural Mechanics competencies give clear guidance on what analysts should know, they were developed by experts in industry, and they are consistent with the other sources reviewed here. Although discretization and matrix representations are indeed two key mathematical abstractions for any FEA professional, analysts use these concepts intuitively. Furthermore, FEA requires numerous practical skills that are absent from typical introductory FEA courses. Analysts scrutinize models by validating against benchmarks and using analysis checks. Students should not only learn to use validation and checks in introductory FEA courses, they should use these tools as a major means of learning FEA itself.

Introduction

Finite element analysis (FEA) is one of the main tools used by structural analysts. FEA is used so often for structural analysis that it is treated as synonymous with computational structural mechanics. Therefore, in this manuscript, FEA will refer specifically to its use in modeling structures, even though the finite element method (FEM) can be applied to other phenomena. FEA is now typically offered as an elective or required course for undergraduate students in mechanical engineering and related disciplines, with advanced courses also available in graduate programs. This paper describes misalignment between introductory FEA courses and industry practice, and uses sources from industry to show how this misalignment can be resolved.

Throughout this paper, we will discuss two major mathematical abstractions that are essential to FEA:

1. *discretization* of continua into elements to convert partial differential equations into algebraic equations, and

2. *matrix representations* of these algebraic equations.

Most people who need to know about FEA are either analysts, or developers of FEA software. There are more analysts than developers, and people taking introductory engineering FEA courses are much more likely to become analysts than developers.

We reviewed the contents of several popular introductory FEA textbooks [1], [2], [3], [4], which incrementally build up the method itself with separate chapters on 1D, 2D, and 3D simulations of single-variable and solid mechanics problems. Textbooks typically use derivations to establish discretization and matrix representations. Textbooks may also devote chapters to related topics such as isoparametric and axisymmetric elements [1], dynamics and optimization [2], numerical integration [3], or shape functions and element formulations [4]; most of these topics are handled with derivations. Based on informal review of syllabi and course descriptions on the internet, most introductory FEA courses for mechanical engineering students, especially in the United States, have a similar theory-focused format, but also include exercises with a commercial FEA tool. Most FEA courses seem designed more for developers than analysts.

Or, perhaps, it is assumed that a theory-first approach is also ideal for analysts, supposing that 1. people need to know how to build solvers to use them responsibly, and 2. fundamental knowledge of numerical methods is sufficient for an analyst to build models. There is no clear evidence for either of these premises.

It is as if academia typically teaches engineers how to make Lego bricks, whereas industry expects engineers to put Lego bricks together. Here, Lego bricks are like elements and studs are like nodes. Someone creating Lego bricks defines the geometry in millimeters, accounts for draft angles, and sizes interference fits, whereas someone designing a Lego structure defines the geometry in terms of studs and plate thicknesses, and uses a variety of bricks to generate a structure of some intended form. The numerous millimeter dimensions for a Lego brick are an implementation detail of no direct relevance to someone building a model, and a model builder is constrained by the abstracted integer stud and plate dimensions, anyway. Critically, integer Lego brick dimensions are an abstraction that enables a builder to focus on the overall model itself.

Likewise, creators of FEA software must derive shape functions and write code that relates loads and displacements at nodes. An FEA analyst does not need to do these tasks, and being preoccupied with these details distracts them from connecting elements to represent structures. Unlike Lego builders, FEA analysts need to comprehend implementation details—namely discretization and matrix representation—but they use these abstractions so differently from FEA developers that it is unlikely that mathematical derivation of these abstractions builds the comprehension that FEA analysts require.

This issue has been noted previously, with Papadopoulos et al. arguing against theory-first approaches to teaching FEA [5], and comparing such approaches to teaching students how calculators work before allowing them to use calculators as engineers do. A book by most of the same authors notes that erroneous use of FEA is rarely due to lack of theoretical knowledge or software defects; in response, they share practical teaching approaches that address misuses of FEA that are made in practice [6].

It is possible to gain clarity on how to teach practical FEA for analysts by reviewing several kinds of sources:

- certification competencies,
- books on FEA for analysts in industry,

- trainings for analysts in industry, and
- solver manuals.

This paper will address each of these types of sources in turn, then describe how analysts use the key abstractions of discretization and matrix representation, describe how analysts check models, and close with recommendations for introductory finite element courses.

Certification competencies

The NAFEMS Professional Simulation Engineer Certification is a credential on simulation engineering that is based on robust industry input. This section will focus on this one certification because it addresses FEA itself and there is nothing else like it; certifications on use of specific FEA products are out of scope for this paper. NAFEMS is a global not-for-profit association supporting the engineering simulation industry [7]; it originated in the United Kingdom as the National Agency for Finite Element Methods and Standards. NAFEMS offers a Professional Simulation Engineer Certification. This credential originated as an EU-funded project, "Certification of COmpetencies in the Power and Pressure Systems Industry" (CCOPPS) with a major focus on the analysis and simulation skills needed in that field [8]. The success of CCOPPS led to a follow-on project, "Engineering Analysis and Simulation Innovation Transfer (squared)" (EASIT2) that went beyond the power and pressure systems industry and focused on skills needed in simulation engineering [9]. Professor Jim Wood of the University of Strathclyde led CCOPPS and EASIT2, and most people involved were engineers in industry. The PSE Certification competencies originated in EASIT2 and are now updated by various NAFEMS working groups appropriate for a given simulation discipline. Certification can be obtained in various technical areas including Computational Structural Mechanics (CSM), which is treated as synonymous with Finite Element Analysis; this discussion focuses on the NAFEMS PSE competencies for CSM. The Computational Structural Mechanics Working Group mostly consists of engineers across numerous industries but also includes consultants and employees of FEA software vendors [10].

Hundreds of engineers hold the PSE Certification, and the uses of the competencies go beyond formal certification. Boeing uses the competency statements themselves for mentoring simulation engineers [11]. At the semiconductor manufacturer ASML, engineers who wish to use CAE software are first required to earn a customized version of the PSE certification [12]. Academic modules (courses and programs) can earn a "NAFEMS Approved" designation by informing NAFEMS of which competencies are covered and explaining how they are assessed [13].

The full set of 87 competencies is available through the PSE Competency Tracker [14]. Competencies are aligned with the six levels on Bloom's taxonomy and designated as "standard" or "advanced;" advanced competency is only required for attaining the highest of three levels of PSE certification—far beyond what is expected of students in introductory courses. Furthermore, for an academic module to earn NAFEMS approval for CSM, NAFEMS only asks about 22 competencies; only four of those are designated as "advanced" and those are mostly qualitative or conceptual. Tables 1 and 2 summarize the CSM competencies at the standard and advanced levels, respectively, and indicate how many competencies are listed for each Bloom's taxonomy level. Some competencies are quoted verbatim.

Table 1: Standard NAFEMS PSE CSM competencies

Level (N)	**Summary of competencies**
Knowledge (11)	Define or list steps in simulation process, degrees of freedom for all common elements, symmetry, forms of element distortion.
Comprehension (18)	Discuss error, checks for accuracy, convergence, displacement versus stress results, shape functions, shear locking, stress averaging. "Discuss the nature of the structural stiffness matrix." "Discuss the relationship between shape function and strain/stress prediction for simple 2D linear and parabolic elements."
Application (10)	Use various elements. Use various boundary conditions. Conduct validation and sensitivity studies, and post-solution checks. "Employ an analysis system for the determination of stresses and strains in small displacement, linear elastic problems."
Analysis (2)	Analyze results and compare them with expected values.
Synthesis (0)	
Evaluation (0)	

Table 2: Advanced NAFEMS PSE CSM competencies

Level (N)	**Summary of competencies**
Knowledge (6)	List or define terminology such as variational principle, degrees of freedom, element types, and required assumptions needed for various idealizations.
Comprehension (22)	Discuss error, matrices (stiffness, conditioning, sparse algorithms), shape functions, element distortion, submodeling. "Explain how the structural stiffness matrix is assembled from the individual element matrices." "Discuss the integral equation for element stiffness, highlighting the variables which it is dependent upon." "Discuss the Geometric Stiffness Matrix and highlight situations where it becomes important."
Application (4)	"Illustrate the various steps in the Displacement Finite Element Method from assumed displacement polynomial to determination of stresses." Use asymmetric and cyclic boundary conditions.
Analysis (1)	Perform sensitivity studies.
Synthesis (8)	Develop analysis specifications and strategy, including benchmarking and validation.
Evaluation (5)	Use and recognize the limitations of idealizations. Manage simulation resources. Do verification and validation.

Figure 1 shows how many CSM competencies deal with discretization and matrix abstractions at each Bloom's taxonomy level and at the standard and advanced certification levels. Overall, for every level, most or all of the competencies do not involve these mathematical abstractions. Most competencies dealing with either discretization or matrix representations are only expected of Advanced analysts. All but 2 competencies dealing with mathematical abstractions are below the Application level and require only qualitative understanding. Such qualitative understanding can be simple and obtained through experimentation with an FEA tool; for example, analysts do need to be able to explain convergence—but you can explain convergence by saying that finer and finer meshes should lead to a stable value, and that value is more and more accurate only if certain assumptions are met.

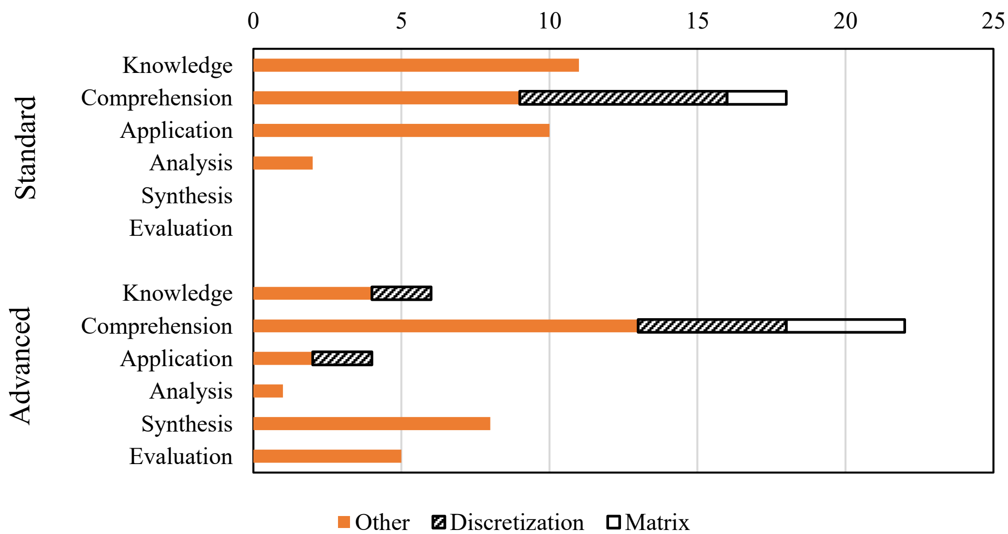

Figure 1: Number of CSM competencies at each Bloom's Taxonomy level, and at the Standard and Advanced levels, for competencies dealing with discretization, matrix representations, and other concepts.

Only two of the CSM competencies directly address actual software use (and then, not any particular package). The remaining competencies focus on responsible use of FEA, but with different techniques and addressing different concerns than are the focus of academic textbooks. FEA models are trusted based on benchmarking studies, which are a major theme of the CSM competencies, especially at the highest Bloom's levels. The CSM competencies additionally include confirming whether the conditions for using an idealization are met, use of boundary conditions, performing quality checks such as element distortion, and avoiding practical flubs like using a mesh with cracks in it.

In summary, the NAFEMS PSE competencies for CSM are a detailed accounting of the skills needed for FEA. Engineers in industry developed these competencies so they reflect engineering practice. Discretization and matrix representations are treated as critical facets of FEA—but at the same time, analysts are expected to address these abstractions conceptually, rather than through derivations. Only advanced analysts—not students—are expected to do derivations involving these concepts. Numerous practical skills for making sound FEA models are included in the CSM competences, which are not emphasized in textbooks.

Books

Some introductory books on FEA are written by and for analysts in industry. The books by Dominique Madier (Ref. [15]), Claudio Gianni (Ref. [16]), and Nitin S. Gokhale et al. (Ref. [17]) will be compared in detail. Madier is an aerospace structural engineer. Gianni works in motorsports (e.g., Formula One) but also has more general experience. Based on their focus on crash simulation and noise, vibration, and harshness (NVH) modeling, Gokhale et al. clearly work in automotive engineering. The contents of these books are summarized in Table 3; items are based on chapter titles, rephrased to show consistency between texts.

These books have similar sets of topics. Meshing, loads and boundary conditions, and connecting bodies or contact between them are treated as significant standalone topics. Various solution types each get a chapter.

These books differ from textbooks. In these books, theory is condensed if it is present. Textbooks are typically structured around the dimensionality of the problem, whereas industry books assume the problem is 3D, and use 1D and 2D elements to represent slender or thin-walled bodies, reserving 3D elements for bulky objects. In industry books, it is assumed that a model will not simply be a single part, but either an assembly of multiple connected or contacting parts, or the model will be a detailed model that requires submodeling. Figure 2 shows an illustrative sample model of an assembly of multiple connected parts, meshed with various element dimensionalities.

Industry FEA books have a philosophy that is consistent with the NAFEMS PSE CSM competencies. Discretization and matrix representations are still the key mathematical abstractions that enable FEA. These books show how analysts use these abstractions conceptually.

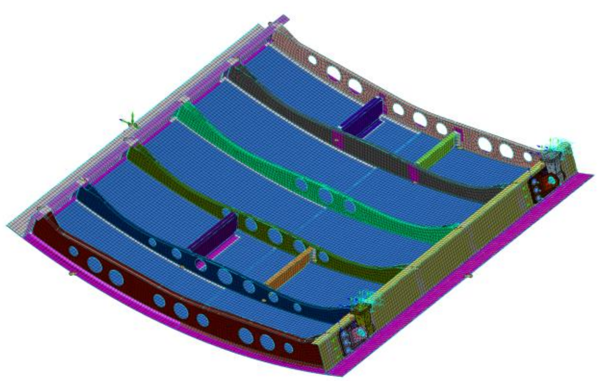

Figure 2: A model of a landing gear door used in industry training on FEA. The skin, ribs, beams, and intercostals are meshed with 2D quadrilateral elements, the tangs and clevises are meshed with 3D tetrahedral elements, and loads are transferred via 1D joint and fastener elements and simulated glue. Image created by the author using MSC Apex.

Table 3: Comparison of FEA industry book contents

Madier	Gianni	Gokhale, et al.
Qualitative description of FEA and careers (4 chapters)	Element types	FEA overview (3 chapters)
FEA theory	Loads and boundary conditions	Mechanics theory
Modeling strategy	Results	Meshing (4 chapters on various element types)
Element types	Modal analysis	Connecting bodies and contact
Meshing	Buckling	Welds and fasteners
Materials	Errors	Materials and boundary conditions
Loads and boundary conditions	Advanced: substructuring and submodeling	Linear static
Couplings (R-type elements)	Composites	Nonlinear
Bolted joints	Validation	Dynamics
Contact	Failure modeling	Thermal
Submodeling	Nonlinearity	Computational Fluid Dynamics
Validation	Dynamics	Fatigue
Results	Optimization	Crash
Performance	Damage	Noise, Vibration, and Harshness
Documentation	Examples	Post-Processing
Linear static	Conclusions	Experimental validation
Nonlinear static	Appendices on structural calculations, stiffness matrix, and numerical linear algebra	Modeling tips and career advice (2 chapters)
Buckling		
Nonlinear buckling		
Normal modes		
Good modeling practices		

Trainings for analysts in industry

Analysts in industry take training on FEA to advance their careers so training must be at least perceived to be relevant to real-world practice. This training is often paid for by employers and knowledgeable managers are often the ones to approve training expenses. Trainings are offered by expert industry FEA analysts. Therefore, it is reasonable to take training content as a sign of the skills that analysts need in industry. We will focus on popular introductory FEA trainings:

- Practical Basic FEA from FEA Academy by Dominque Madier and Steffan Evans [18],
- Breakthrough FEA Course from Enterfea by Łukasz Skotny [19], and
- Basic Finite Element Analysis from NAFEMS by Tony Abbey [20].

These courses are taught by consultants who have done FEA analysis for numerous organizations, with Madier, Evans, and Abbey mostly in aerospace structures, and Skotny in civil engineering structures.

In the Practical Basic FEA course, about 75% of the course is conceptual and is vendor-neutral; the remaining 25% addresses corresponding software workflows. The other courses are vendor-neutral.

Madier and Evan's course follows a similar outline to Madier's book, which was described in Table 3; the course includes the mathematics of discretization and matrix representations but spends only about 6% of its time on these topics. Skotny's course addresses structural mechanics theory, model building, connecting different element types, boundary conditions and contact, loads, meshing, various analysis types, and post-processing. It does not derive mathematical abstractions, but the topic of discretization is handled practically by comparing element types and offering advice on stress averaging. Abbey's course, being offered through NAFEMS, specifically addresses many PSE CSM competencies; the course outline includes stiffness matrices, shape functions, element types, sanity checks, convergence, stress raisers, loads and boundary conditions, analysis checks, simplifications, and interpreting results. All emphasize model checks (e.g., element quality or making sure stress fields are well-resolved) and validation against benchmarks.

Madier lists statics and solid mechanics, but not any mathematics courses, as prerequisites. Skotny's and Abbey's courses are designed primarily for working engineers and do not otherwise have prerequisites. There is no indication that any theoretical coursework on FEA be completed before taking one of these courses.

Taken together, the structure of these trainings reinforces the main points shown in the previous section on industry books.

Solver manuals

Analysts mostly use commercial FEA packages. Ansys Mechanical, Abaqus, and various flavors of Nastran are the primary general-purpose solvers, not counting CAD-integrated packages such as SolidWorks Simulation. Although preprocessor user interfaces might conceal the details of the analysis, analysts who demand high performance and reliable results turn to solver manuals to find essential details. Solver manuals are typically published as many volumes, totaling tens of thousands of pages. They do not explain how to create a solver but instead enable analysts to connect their theoretical knowledge of FEA to solver inputs, and show how to interpret results in light of the idealizations used by the solver. Here, we compare how knowledge is structured in the manuals for Ansys Mechanical APDL, Abaqus, and MSC Nastran, and discuss the specific topics of shape functions and stiffness matrices.

Solver manuals tend to organize knowledge based on steps in the analysis process. The front matter of the Abaqus manual encourages the learner to first do some exercises but then to read separate volumes on "Modeling Capabilities & Theory" which mostly correspond to the steps in the process [21]. Taken together, the Ansys Modeling and Meshing, and Basic Analysis guides cover the process. The first half of the MSC Nastran Linear Static Analysis User's Guide [22] is basically organized by the steps in the process, with the second half largely covering advanced or niche topics.

Solver manuals are written with the expectation that the reader has heard of shape functions, but they do not mention them much and do not expect most readers to do much with them. If a manual even shows a derivation of shape functions, the explanation is much terser than is typical for a textbook. The shape functions themselves may not be given, notably because structural elements such as beams and shells can be formulated in various ways and exact formulations are sometimes trade secrets. Abaqus [23], [24], and Ansys Mechanical [25], [26] each have separate manuals on background theory and element details, with shape functions being discussed or given in the theory manuals. The Abaqus theory manual [23] mostly uses shape functions to explain how advanced or unusual elements work, or to describe the angle of orthotropic materials. The Ansys theory manual [25] generously shares a chapter of isoparametric shape functions, but with little explanation or context for them. The MSC Nastran reference manual [27] is written with the assumption that the reader knows what a shape function is, but it only shows shape functions for one unusual type of shell element; shape functions are also used to describe how mesh refinement splits are made in isoparametric space. Apparently, shape functions would not be germane to choosing among elements in a catalog; instead, analysts choose elements based on dimensionality, shape, and order.

Solver manuals are written with the assumption that all users are aware that stiffness equations are represented in a matrix form. Matrix terminology is used even in getting started manuals [22], [28], [29]. Two vendors give detailed examples with two rod elements in series, showing the contents of a stiffness matrix along the way [22], [28]. At the same time, most users are not expected to use mathematical notation for matrices. Different solution types (e.g., linear static, buckling, nonlinear) are explained in terms of the matrix equations that are solved but such explanations are often given in theory manuals written for advanced users [23], [25], [27]. Analysts may need to pick between matrix solvers, but they are typically given a table that compares solvers qualitatively based on their typical uses, pros, and cons. Manuals mention matrix methods, such as Cholesky decomposition or the conjugate gradient method; those labels are beneficial for users who are aware of advanced linear algebra methods.

In summary, solver manuals, like the other sources discussed so far, structure knowledge in terms of steps of the analysis process. Although discretization and matrix representations are addressed, most readers are expected to use these concepts intuitively. Wherever possible, choices are also framed in terms of the assumptions that are required for them to be valid, or their practical consequences. Occasionally, more details are offered to advanced users on unusual fine points; even then, readers are rarely expected to follow derivations.

How analysts use discretization

This section synthesizes the sources mentioned so far to explain how analysts use discretization, given that they are not performing derivations and writing solvers themselves. (The next section will do likewise for matrix representations.)

Analysts use conceptual understandings of discretization throughout the analysis process. Choices are dictated by analysis goal, with stress analysis requiring fine meshes, explicit solutions generally requiring linear elements, and other analysis types being more flexible.

Analysts make most choices in terms of guidelines rather than mathematical details; companies set internal benchmarks on mesh density near stress raisers.

Element dimensionality is typically based on whether a part is slender, thin-walled, or bulky. Elements often correspond to fabrication process, with 1D elements for extrusions, 2D elements for sheet metal, and 3D elements for castings and machined parts. Analysts distinguish between "continuum" behavior of elements that arises from discretization, or "structural" behavior of beam and shell elements that is based on mechanics theory; structural behavior can be very accurate and economical. Linear triangular and tetrahedral elements have poor performance but linear quadrilateral and hexahedral elements have good performance; although you could use derivations to explain why that is, or simple benchmarking to establish these facts, an analyst only has to follow simple guidelines to pick element order based on element shape. Despite the numerical benefits of hexahedral and quadrilateral elements, obtaining meshes with these elements often requires defeaturing and segmenting the geometry; in contrast, almost anything can be rapidly and automatically meshed with quadratic triangles and tetrahedra, and that is sufficient for small models. Therefore, element shape (and thus order) depend on how meshable a part is and the balance of available analyst and computer time.

Meshing takes judgment, especially for performance-intensive models. Analysts use intuition about structures to anticipate which regions will have the highest stress and mesh accordingly. Rules of thumb are often used, with different expectations if stresses are to be extracted and if stress singularities are acceptable: for example, an analyst might use three elements across a fillet to extract stresses, one element to prevent stress singularity, or allow a sharp internal angle (but know to ignore the high stress at that location). Convergence studies on full models may be impractical but convergence studies on small check-out models can be used to develop rules of thumb. Often, analysts rely on element quality metrics (such as aspect ratio, skew, and Jacobian ratio), typically using an aggregate of several metrics.

Boundary conditions must be discretized. Loads and constraints are expressed in terms of degrees of freedom, with different element types having different numbers of degrees of freedom per node. An analyst applying a force usually intends to target a face of a solid or an edge of a surface; however, the stiffness equations are written in terms of nodes. Equivalent nodal loads on elements higher than linear order have non-obvious weightings, and corner nodes may even have loading directions reversed from the overall load. Discretization of loads may be done by the preprocessor, the solver, or the analyst—the analyst must be aware of this issue. Applying a moment is typically done with what Ansys and Abaqus call couplings and what Nastran calls R-type elements (namely RBE2 and RBE3). Naïve approaches to constraints can over stiffen the model and have stress results compromised by Poisson effect artifacts—such approaches may be acceptable if postprocessing is done conscientiously.

When interpreting stress results, analysts should be aware that the solver is generally calculating stresses at Gauss points within the element and extrapolating those stresses to the nodes; then, the stress displayed at a node is averaged from the nodal stresses calculated for each element touching that node. To check results for reasonableness, analysts should disable this averaging in post processing and see if the stress gradients in adjacent elements appear continuous; if stress gradients at adjacent elements have opposing directions, the stress results near those elements are not reliable. If results check out, analysts can then use the averaged nodal stresses.

Analysts should be able to perform global and local convergence studies, expect smaller elements to be more likely to give converged results, and have reasonable expectations of rate of convergence. They should also expect divergence at stress singularities and use other methods to predict stress values at those locations.

In summary, analysts must use concepts about discretization throughout the analysis process, intuitively or with rules of thumb based on experience. Figure 3 shows a schematic of a finite element with essential features indicated. If an analyst thinks of a finite element as an elastic object with

- forces transferred at nodes,
- displacement varying linearly or quadratically (depending on order) within the element,
- element boundaries kept as lines or parabolas, and
- stresses calculated at discrete Gauss points within the element,

and the analyst can intuit the consequences of these idealizations, that is often sufficient for the analyst to make responsible engineering judgments about discretization.

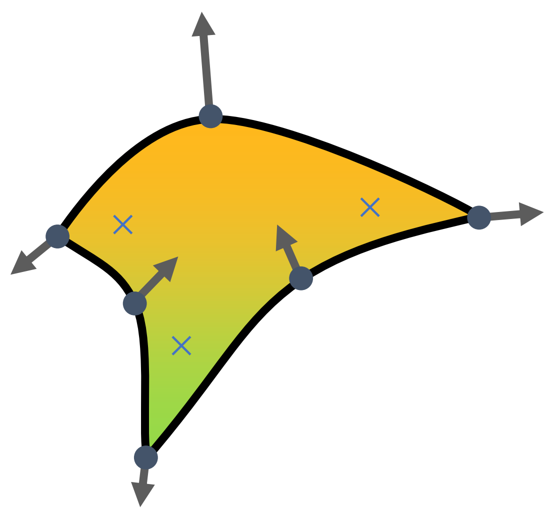

Figure 3: Schematic of a second-order triangular finite element, showing forces acting at nodes, a displacement gradient, parabolic element edges, and Gauss points where stresses are actually calculated.

How analysts use matrix representations

It might seem that analysts do not need to know much about matrices. Unlike discretization, matrix representation mostly does not involve engineering judgments about idealizing a physical system. It is rare for an analyst to even look at the stiffness matrix for a real problem. Instead, matrix representations enable fast computer solutions of large problems. Analysts need to know about stiffness matrices, and methods for solving systems of linear equations and eigenvalue problems because they need to be able to,

- Account for the limitations of linearization.
- Reason about when degrees of freedom are eliminated (constraint equations are condensed out early in the solution process).
- Relate mesh connectivity, stiffness matrix density, and performance.
- Pick solvers and solver settings.
- Manage performance and numerical accuracy.
- Interpret error messages that use matrix terminology.

Still, because analysts do not manually assemble a global stiffness matrix from elemental stiffness matrices, devise matrix algorithms, or write solvers, it is not efficient for them to learn these topics.

It is awkward to align curriculum with the skills used by analysts because although a competent analyst might never use skills from an introductory linear algebra course to manually solve a system of equations or calculate eigenvalues and eigenvectors, solvers typically use techniques that are far beyond such introductory linear algebra courses, namely sparse representations and solution methods like Cholesky decomposition or the conjugate gradient method. Most analysts are not taking coursework that teaches how to derive or program these advanced techniques.

Most of the essential skills above could be handled by teaching vocabulary, or giving plain-language explanations like,

- If a system of linear equations does not have as many equations as unknowns, it is not solvable and we would call it "singular." If you under constrain a model and allow rigid-body motion, the stiffness matrix is singular.
- A system could be close to singular. We measure how singular a matrix is with a "condition number." An ill-conditioned matrix has a condition number much greater than 1 and would be likely to give inaccurate results.

Most analysts can simply use the default solver settings. For tough problems, such as large models or models with highly connected meshes, manuals have guidance, often in tables, on which solver and settings to use; these tables could be discussed in class. Assignments could call for students to attempt tough problems with different solver settings and compare performance (solve time and memory) and convergence.

Solvers typically give an error when a static solution is sought but the stiffness matrix is singular. Stiffness matrix singularities are often caused by physical instabilities, but solver error messages typically do not hint at the root cause of the singularity. Instead, analysts are expected to know that a singularity could be caused by over or under constraint, misapplied contact, invalid material properties, buckling, or hourglass modes (which are non-physical) [29].

How analysts scrutinize models

This paper has shared many industry perspectives on teaching finite element analysis. Having compared how various topics are covered in industry training versus academic coursework, we risk obscuring a critical point: the most important difference between industry and academic mindsets is how we gain confidence in models. Traditional coursework is based on the mathematical proofs used to develop finite element codes. (This approach is indeed essential for developing solvers.)

In contrast, engineers in industry normally build models by relating their problems to benchmarks. Suppose an engineer needs to model a plate with two holes in it that is held in tension. They could develop a benchmark model with one hole that they could compare to a stress concentration formula. They would need to refine the mesh, and could repeat this process until they get a good match to the formula. Then, they could make a mesh with similar density on the plate with two holes. Naturally, to be professionally prepared, students should engage in such exercises so that they learn how to validate and verify real models, which is at the highest Bloom's taxonomy level in the NAFEMS PSE Competencies.

Of course, validation is done against a benchmark that is only similar to the analysis problem, because the analysis problem does not have a known solution. Therefore, additional checks are needed, partly to confirm that the benchmark indeed resembles the problem, and partly to root out mistakes in model building. Pre-analysis checks include,

- Soundness of geometry used to generate mesh
- Element quality
- Meshing errors, such as duplicated elements and mesh cracks
- Fit of assemblies, especially those including structural elements
- Double-checking input values
- Mock normal modes runs to visualize constraints and connections

Post-analysis checks are also done, including,

- Reasonableness of deformed shapes
- Error metrics such as stiffness matrix condition number, and magnitude of residuals
- Comparison of reaction loads to applied loads (for static solutions)
- Applicability of assumptions (e.g., for linear solutions, whether the results are compromised by large deflections or stresses beyond the yield point)
- Smoothness of stress fields (show unaveraged stresses)
- Reasonableness of results (compared to rough estimates)

Convergence studies are frequently used in introductory courses but must be used sparingly and with care in industry. Stress singularities prevent convergence and poor modeling choices (e.g., sloppy boundary conditions) can cause convergence to incorrect values; therefore, convergence is not sufficient for showing accuracy. Global convergence studies are expensive and are rarely feasible on real problems. Local adaptive mesh refinement can get around some of these problems if performance permits. More often, analysis teams perform convergence studies on small "check-out" models to build a library of best practices. For example, holes would be meshed with a minimum number of elements, and that minimum number would vary based on hole size and whether the hole is in a critical region. Therefore, the example above with the plate with two holes, is a simplification: the geometry with two holes would be meshed not based on a one-off convergence study but based on a general guideline on holes.

The experiential, inductive approach of building a finite element model by comparison with benchmarks and using checks better emulates how most people learn, compared with the derivation-focused deductive approach that has been traditionally used to teach the subject. Thus, this approach is not just more industrially relevant, it is more pedagogically sound.

Conclusion

Compared with introductory FEA courses, analysts use theory less formally and, at the same time, use solvers built with methods that are more advanced. Analysts are not deriving discretizations but they are choosing between multiple element types, picking options that affect element formulation, and even using element formulations that are trade secrets; they use intuition about discretization to create non-uniform meshes, connect multiple bodies, apply loads and constraints, and interpret results. Because matrix representations have minimal direct connection to engineering judgment, analysts mainly use concepts about matrices to pick solver options (for performance) and to troubleshoot singular matrices (typically due to under constraint).

Analysts are not writing matrix solvers, but the solvers they use are based on advanced methods like Cholesky decomposition or the conjugate gradient method, and employ sparse methods and high-performance optimizations. It is rare for analysts to even look at a stiffness matrix and viewing shape functions is generally not even possible. Therefore, it is implausible that knowledge gained through derivations or coding exercises (on simple FEA solvers) transfers to the capabilities needed for structural analysis. The NAFEMS PSE CSM competencies focus on discretization and matrix representations at the lowest two levels of Bloom's taxonomy, which indicates that analysts need to have only a conversational knowledge of these subjects. Of course, developers need procedural knowledge of these mathematical abstractions, as well; in contrast, analysts need procedural knowledge of using FEA to analyze structures.

Based on industry perspective on how analysts learn and use FEA, introductory FEA courses for analysts should be modified according to the following recommendations:

- Address mathematical abstractions of discretization and matrix representations conceptually, focusing on the procedural skills that analysts use to manage these abstractions.
- Structure knowledge in terms of steps in the analysis process, shown in Table 4. This is not a suggested course sequence, but when a student learns a concept they must be aware of which step in the process it is relevant to.
- Present certain skills as standalone topics; sources point especially to loads and boundary conditions, connecting multiple bodies, and post-processing.
- Scrutinize finite element models primarily through checks and benchmarking.

Table 4: Steps in the FEA process

Analysis planning
Element selection
Meshing
Loads and boundary conditions
Couplings (R-type elements)
Fasteners and contact
Materials
Solving
Post-processing
Pre- and post-analysis checks
Validation against benchmarks

Conflict of Interest Statement

The author of this paper works at MSC Software (now a part of Cadence Design Systems), which develops engineering simulation software. This paper does not endorse any product. The author became a member of the NAFEMS CSM working group after this manuscript was submitted.

Acknowledgements

The author is grateful to Alex Reichanadter for helpful discussions and a thoughtful suggestions on the manuscript.

References

[1] Daryl L. Logan, *A First Course in the Finite Element Method*, 6th ed. Cengage Learning, 2016.

[2] Saeed Moaveni, *Finite Element Analysis: Theory and Application with ANSYS*, 5th ed. Pearson, 2020.

[3] J. N. Reddy, *Introduction to the Finite Element Method*. McGraw-Hill Education, 2019.

[4] O. C. Zienkiewicz, R. L. Taylor, and S. Govindjee, *The Finite Element Method: Its Basis and Fundamentals*, 8th ed. Butterworth-Heinemann, 2024.

[5] J. M. Papadopoulos, C. Papadopoulos, and V. C. Prantil, “A Philosophy of Integrating FEA Practice throughout the Undergraduate CE/ME Curriculum,” in *2011 ASEE Annual Conference & Exposition*, 2011.

[6] V. C. Prantil, C. Papadopoulos, and P. D. Gessler, *Lying by Approximation: The Truth about Finite Element Analysis*. Morgan & Claypool Publishers, 2013.

[7] NAFEMS, “About Us.” Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/about-us/

[8] NAFEMS, “Competencies in the Power and Pressure Systems Industry.” Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/about-us/projects/past-projects/ccopps/

[9] NAFEMS, “Engineering Analysis and Simulation Innovation Transfer.” Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/about-us/projects/past-projects/easit2/

[10] NAFEMS, “Computational Structural Mechanics Working Group.” Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/community/working-groups/computational-structural-mechanics/members/

[11] NAFEMS, “PSE: What’s In It For Me?” Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/events/nafems/2025/professional-simulation-engineer-certification-pse-benefits/

[12] I. Symington, “NAFEMS Professional Simulation Engineer: An Industrial Case Study.”

[13] NAFEMS, “Course Accreditation & Module Approval,” NAFEMS. Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/community/working-groups/education-and-training/accreditation_and_approval/

[14] NAFEMS, “PSE Competency Tracker.” [Online]. Available: https://www.nafems.org/professional-development/competency_tracker/

[15] D. Madier, *Practical Finite Element Analysis for Mechanical Engineers*. FEA Academy, 2020.

[16] Claudio Gianni, *Computational Structural Engineering*. CG CAE Sagl, 2022.

[17] Nitin S Gokhale and S. V. B. Anand N Thite Sanjay S Deshpande, *Practical Finite Element Analysis*. Finite To Infinite, 2008.

[18] “Practical Basic Finite Element Analysis,” FEA Academy. Accessed: Jan. 14, 2026. [Online]. Available: https://fea-academy-learn.thinkific.com/courses/basic-fea

[19] “Breakthrough FEA Course,” Enterfea. Accessed: Jan. 14, 2026. [Online]. Available: https://enterfea.com/breakthroughfea/

[20] NAFEMS, “Basic Finite Element Analysis course.” Accessed: Jan. 14, 2026. [Online]. Available: https://www.nafems.org/training/e-learning/basic-fea/

[21] Dassault Systèmes Americas Corp., “Abaqus: Front Matter,” 2026, 2026.

[22] Hexagon AB, “MSC Nastran: Linear Static Analysis User’s Guide,” 2025.1, 2025.

[23] Dassault Systèmes Americas Corp., “Abaqus Theory Guide,” 2026, 2026.

[24] Dassault Systèmes Americas Corp., “Abaqus Elements Guide,” 2026, 2026.

[25] ANSYS, Inc., “Mechanical APDL: Theory Reference,” 2025 R2, 2025.

[26] ANSYS, Inc., “Mechanical APDL: Element Reference,” 2025 R2, 2025.

[27] Hexagon AB, “MSC Nastran: Reference Guide,” 2025.1, 2025.

[28] Dassault Systèmes Americas Corp., “Getting Started with Abaqus/CAE,” 2026, 2026.

[29] ANSYS, Inc., “Mechanical APDL: Basic Analysis Guide,” 2025 R2, 2025.