A Concern over Computer-Aided Engineering

by Kolbjorn Saether

Historically, the engineering of structures began with intuitive notions of stability and strength of materials which gave rise to Egyptian pyramids starting in the 3rd millennium b.c. and followed with notable Persian, Greek and Roman constructions that included temples, coliseums, and aqueducts. During the Renaissance, the elucidation of the fundamental scientific principles of rigid body mechanics, elastic material behavior, and mathematical formulations of common structural components such as plates and columns took place. Modern developments include the availability of steel and reinforced concrete from which highrise structures could be built.

The desire to build highrise structures necessitated the development of analysis techniques using an assemblage of subcomponent frames, columns, and plates for which rigorous mathematical formulations could be utilized for their analysis. These components could be accurately analyzed to determine sizes and reinforcement to carry live, dead, wind and seismic loads. With the advent of computers, these analysis methods were the first to be programmed into software programs and led to tremendous reductions in design cycle times and, concomitantly, reduced errors resulting from hand calculations. The development of computer platforms and specialized software programs have revolutionized the engineering field and will continue to extend the detail to which structural designs can be analyzed to assess critical deflection and strength behavior under assumed service loads.

General numerical procedures such as finite difference methods were supplanted in the mid 1960's by the development of the finite element method. Finite element techniques provide and analysis paradigm ideally suited for computer implementation. This method is applied by discretizing a continuous structure into a field of nodes at which translations and rotations are assumed as unknown discrete variables. Finite elements are defined by a connected sequence of nodes and assume a simple variation of field quantities - such as displacements - over the domain of the element. Minimizing the elastic energy within the element domain with respect to arbitrary variations in the nodal variables yield the element stiffness characteristics. These elements may be one, two, or three-dimensional to represent beams, plates, and solids. A complete structure is then represented by a connected assemblage of finite elements with loading and support conditions applied to nodes. This technique replaces the use of continuous differential equations for the analysis of separate subcomponents with a discrete representation of the structure in the form of a large system of algebraic equations which can be rapidly solved on a computer using techniques of matrix algebra. The fundamental equation to be solved is written as:

$$[K]\{u\} = \{f\} \qquad (1)$$

where [K] is the matrix of stiffness coefficients, {u} is the vector of unknown displacements and rotations, and {f} is the vector of applied forces. This computational paradigm has enabled the practical implementation of algorithms to perform linear and nonlinear static and dynamic analysis, design optimization, progressive failure analysis, and nondeterministic probabilistic analysis.

General commercial finite element codes such as NASTRAN, ABAQUS, and ANSYS, are widely used in the aerospace, automotive, and shipbuilding industries to analyze new design concepts. The need to create large finite element models has lead to the introduction of powerful commercial pre- and postprocessing programs such as PATRAN and IDEAS to provide a graphical environment to construct model geometry, create finite element meshes, and to display analysis results. To further simplify the creation and solution of large finite element models, every field of engineering has available a competing suite of specialty software that are specifically designed to construct specific models. For example, the specialty finite element analysis program ETABS, written for the structural engineering field, is quickly evolving into an industry standard through its widespread use.

Of great concern is the gradual separation of the structural engineer from working directly with the rigorous governing equations describing the behavior of slabs, columns, and shear walls. Instead the engineer is required to deal with computer models requiring tens of thousands of nodes and elements and resulting in a discrete solution that could be in the order of hundreds of thousand nodal displacements. This appears to lead naturally to a disconnect in the ability of the engineer to develop an engineering judgment or "feel" to identify incorrect solutions. The chance of introducing errors into a finite element model that go unnoticed is profound, particularly in specialty codes that keep more of the model construction hidden by providing graphical interfaces for preprocessing. These interfaces attempt to simplify model development by allowing large subcomponents to be selected and incorporated into the model through menuing schemes. This is particularly risky for professionals that are not skilled in numerical methods who may judge the validity of a solution partly on the reputation of the software or be prone to what is commonly refereed to in industry as `NASTRAN Fundamentalism" in which the solution output, generated to eight significant digits and the crisp output formatting of tables and graphics, can put the results into an irrefutable light and is trusted a priori. In other words, the presentation of the output can often create a belief in a solution which precludes an unbiased scrutiny. Specifically, if errors are present in the structural model that do not prevent an analysis from being performed, the finite element analysis will give the absolutely correct solution to the definitely wrong problem.

In light of recent developments in computer software in the field of structural engineering such as SAP and ETAB a warning seems appropriate. The amount of sophistication built into these programs necessarily obscure to the design engineer the true structural behavior and the potentials of the project. For example, this played out in an ugly fashion on a recent 60-story highrise project. This project entailed a 3 story parking garage with two 60-story towers perched on top. The project was carefully engineered and was given a peer review by an outside engineering office with the full cooperation of both the architect and the lead engineer on record. The peer review firm utilized ETABS to model the 60-story tower and showed a deflection three times the allowable. This caused the architect and the owner to panic and they both demanded an explanation from the lead engineer. The lead engineer, with over 50 years of professional experience in highrise design, had already developed a detailed engineering plan and construction had proceeded to the beginning of typical tower floors. After obtaining a copy of the ETAB model, the lead engineer hired an ETABS consultant who was able to pinpoint serious errors in the model. A major error involved the omission of two entire floor slabs that was easily hidden in the automated model generation routines. Once corrected, it was found that the deflections were exactly as the original design had predicted, one third of what had been forecast by the faulty ETAB model and exactly what was allowed by the industry.

In the meantime however, due to the scare of a design deficiency and delay in production of the structural drawings, the architect/owner fired the original engineer, and put the peer review firm in charge to finish the structural design of the project. By this time the foundation had been completely installed. The new engineering firm followed the findings generated by the faulty ETAB model, changed the engineering concept and produced a new set of structural drawings. In this transfer a serious slip-up took place. The original layout had included a transfer of wind-load at the 10th floor of the 60-story tower. At this level the load was to be distributed over a greater number of columns resulting in little or no increase in column and caisson sizes and no addition of tension steel in the caissons. The new layout did not accomplish this load-distribution and severe shortages in the existing foundation developed.

Another example was luckily interrupted in its design stage. When initially analyzed by an ETAB's model, an eccentrically located elevator shaft caused torsional moments to dominate the design of elevator core walls. The result of the analysis indicated the need for 22 inch thick shaft-walls carried from ground floor up to the penthouse of this 33 story building. By introducing the most distant end-wall into the overall model and by giving the wall a slight modification in the layout, the eccentric shaft-load in the form of torsion was eliminated and 12 inch walls resulted.

The two examples above describe different problematic issues when using advanced computer programs. The first is inherent to the ease-of-use capabilities built into these specialty codes that necessarily hides many aspects of constructing a finite element model. A parallel effort is advocated to develop better internal error detection that may involve artificial intelligence methodologies such as expert systems that could encode in some fashion expert knowledge to automatically assess developed models for both obvious and subtle omissions or incorrect model construction. The second issue pertains to the diminished experience developed by engineers in their understanding of the complexity of the structures when exclusively using advanced computer programs.

As a remedy to the second issue, the engineer on record should be required to develop the ability to create short computer routines allowing the creation of sound, approximate values against which the refined computer generated structural solutions can be evaluated. These approximate analyses would furnish the engineer an insight into various structural systems and thereby allow him to develop the necessary physical insight to arrive at optimum solutions. As demonstrated in the second example above, a slight modification of one part of the design eliminated a situation in which other parts of the structure, the elevator shaftt thickness, were clearly notoptimized.

It is this writer's opinion that, without this added set of analyses, the complete dependency of complex structural computer models could be of questionable aid to the structural profession.

Several methods are available to eliminate or minimize the above described shortcomings:

1) ***First of all, the engineer must be thoroughly familiar with any advanced program he is using***.

2) ***If the program has graphics capabilities the engineer should make full use of getting a visual picture of the complex computer model.***

3) ***The third, and perhaps the most important step, by the use of short routines using approximate and average values with simple calculations the engineer should get an estimate for the overall loads, reactions and deformations of the structure.***

After the base behavior has been established with these routines, a complex finite element model could be validated and safely be used to refine and expand the initially established values